EVALUATION OF VIRTUAL PRIVATE NETWORK IMPACT ON NETWORK PERFORMANCE

By

MUKATSHUNG CLAUDE NAWEJ

Submitted in accordance with the requirements for
the degree of

MAGISTER TECHNOLOGIAE
Electrical Engineering (Computer systems)

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: PROF SHENGZHI DU
CO-SUPERVISOR: PROF FRANCOIS MULENGA

September 2016

Abstract

The aim of the study is to investigate what impact the use of VPN has on network performance. An empirical investigation using quantitative research methods was carried out. Two sample scenarios were involved in the study: scenario without VPN and scenario with VPN. In both scenarios, three applications were used in turns, an HTTP, an FTP, and a CBR. FTP was configured to use window size and packet size, while CBR used connection rate and packet size. On the other side, the number of connection was the only parameter used for HTTP. These applications were injected in a 100 Mbps fixed link of an NS2 simulation environment. Throughput and delay averages were measured respectively for the two scenarios and values compared using Student's t-test.

While the TCP and HTTP throughputs were found decreasing, the UDP throughput was not affected by the presence of this VPN. Concerning the delay; the TCP, UDP and HTTP delay were found increasing.

Key words:
Simulation, VPN, NS2, Network performance

Declaration

Student number: **5085-409-7**

I declare that the **EVALUATION OF VIRTUAL PRIVATE NETWORK IMPACT ON NETWORK PERFORMANCE** is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I have not previously submitted this work, or part of it, for examination at UNISA for another qualification or at any other higher education institution.

---------- ----------

Signature Date

(Mukatshung Claude Nawej)

Acknowledgements

Firstly, I would like to thank the One and only one pillar of my strength, God, the eternal rock of ages for giving me the strength to complete this dissertation. Secondly, I thank my supervisors, Professors Shengzhi Du and Francois Mulenga whose help and stimulating suggestions guided me during my research and write-up of this dissertation.

Next, special thanks go to my beloved kids Eden Nawej and Dan Nawej for enduring sacrifices which led to this achievement. Special acknowledgements to my family for their constant love, prayers and appreciation to my desire to go for further studies.

Finally, I would like to show my deepest gratitude towards my old and childhood friend Professor Lucks Lukanda Kalobo for his constant encouragement, motivation and help towards building my confidence.

Table of Contents

List of Figures

List of Tables

List of abbreviations and acronyms

CBR	: Constant Bit Rate
FTP	: File Transfer Protocol
HTTP	: Hypertext Transfer Protocol
IP	: Internet Protocol
ISO	: International Organization for Standardization
J-SIM	: Java Sim
LAN	: Local Area Network
MAN	: Metropolitan Area Network
MPLS	: Multiprotocol Label Switching
NS2	: Network Simulator version 2
OSI	: Open System Interconnection
OTCL	: Object Oriented Tool Command Line
QoS	: Quality of Service
TCL	: Tool Command Language
TCP	: Transport Control Protocol
UDP	: Data Protocol
VPN	: Virtual Private Network
WAN	: Wide Area Network
WLAN	: Wireless Local Area Network

Chapter 1: Introduction

1.1 Background and motivation

A virtual private network (VPN) is a network device or series of devices used to interconnect various geographically separated sites, or connect remote users back to a home network. As VPNs continue to evolve with a growing number of options, they become more attractive to small business owners in terms of providing secured communication over a public infrastructure like the internet (Narayan *et al.*, 2009). VPNs also offer benefits such as flexibility, connectivity and security at a cheap cost. Enterprises benefit from VPN in reducing the cost, increasing the scalability, and increasing the productivity without compromising the security (Fisli, 2005).

The aim of this present research is to investigate the performance of networks using Network Simulator version 2 (NS2), a network simulation environment. Two setups are considered where in the first instance the network is without VPN, and in the second the network is coupled with a VPN. Per scenario, various trigging parameters are changed alternatively in the simulation. The computed and calculated throughput and delay of some common network applications like Constant Bit Rate (CBR), File Transfer Protocol (FTP), and Hypertext Transfer Protocol (HTTP) are assessed using statistical analysis. The assessment's results of these two scenarios are compared to interpret the impact.

1.2 Problem statement

In the current era of information, VPN plays a crucial role in the sense that productivity is not compromised as users can get connected from anywhere the internet. However, it has been noticed practically that communication with and without VPN are different with the former being slow. This weakness has brought about the need to establish the impact of a VPN on the network performance.

1.3 Research objectives

From the previous section, it is understood that in terms of performance, a VPN-based communication system is relatively slow. It is the purpose of this research to determine the impact and extent of VPN usage on network performance. The specific objectives are:

- To identify common network performance metrics and evaluate its VPN's impact
- To compare the VPN's impact from different common network applications perspective
- To recommend on how to reduce this VPN network performance impact

1.4 Research Questions

Based on the above research objectives and considering that the most common network performance metrics are throughput and delay (Hasib, 2006), the work seeks to answer the following questions:

- What is the common network performance metrics, do VPNs impact them?
- Does VPN's impact vary differently with a different type of applications or protocols?
- What needs to be done in order to reduce the identified impact?

1.5 Scope of the study

The research study is simulation-based; as such, no real life testing or implementation is done. It has not been possible to run a test bed experiment on a specific network due to logistical limitations.

The simulations are set to cover three applications: HTTP, FTP and CBR. In terms of traffic settings parameters, packet size, window size, transfer rate, and a

number of connections are used. And for all simulations, the default bandwidth of 100 Mbps is used to mimic as closely as possible to real-world networks.

1.6 Layout of the dissertation

This dissertation comprises seven chapters. Chapter one presents the background to the study, problem statement, objectives, research questions followed by the scope of the research, and the structure of the dissertation.

In Chapter two, a review of relevant studies to the topic is done. The chapter introduces the importance of network layer structures. Topics relating to VPNs, network simulation, and network performances are covered.

Chapter three describes in detail the research methodology, procedures and techniques used in this dissertation. Chapters four through six deals respectively with the statistical analysis and interpretation of the CBR, FTP and HTTP simulation data gathered.

Finally, chapter seven conclude the dissertation by summarising the relevant findings. Future work is also identified and recommendations are made in line with this.

Chapter 2: Literature review

2.1 Introduction

This chapter gives an overview of the research and the prior research studies that have been done regarding the research stream. It considers what questions have been addressed and answered before, and what results have been produced. For a strong foundation and basis for this research, some definitions and the network reference model networking concept are presented first. This is followed by the main component of the research, network performance. Furthermore, to identify the research gaps in order to develop the research study, an overview of the current trends in research and industry related to the VPN and its impacts is discussed. Literature was collected from credible resources like journals, research reports, books and internet searches. The review of literature starts with the next section which is the definition of concepts.

2.2 Clarification of concepts

Since the research focus is mainly on the term VPN, it is fair enough that research starts by defining this term. VPN has come to mean many things to many people. There are thousands of books that define it in whichever way serves the context.

For simplicity of this research, the VPN is defined as a network device or series of devices used to interconnect various geographically separated sites, or connect remote users back to a home network (Tan, 2003). To accomplish this functionality, the VPN server uses two network interface cards. One connected to the intranet network and another one connected to the public network. Staff connected to the internet, find a VPN server, and then connect to the corporate intranet. VPN actually uses a common link on the internet and it is essentially the use of encryption technology to encapsulate a data communication tunnel which transmits data on the public internet. The internet has a characteristic that it is structured with a uniformed network stack (TCP/IP) that all the different layers technologies can be implemented differently but with a uniformed interface with

their neighboured layers, Open System Interconnection Reference Model. The next section explains the reference model.

2.3 Open system interconnection

At the beginning of the internet, there were many communication frameworks. The result was if one's framework is chosen, then the corresponding product should be the only one with the same framework for network connection to happen. Fortunately, the International Organization for Standardization (ISO) published the Open System Interconnection reference model (OSI). It is a layered structure aimed to provide a framework of open system protocol.

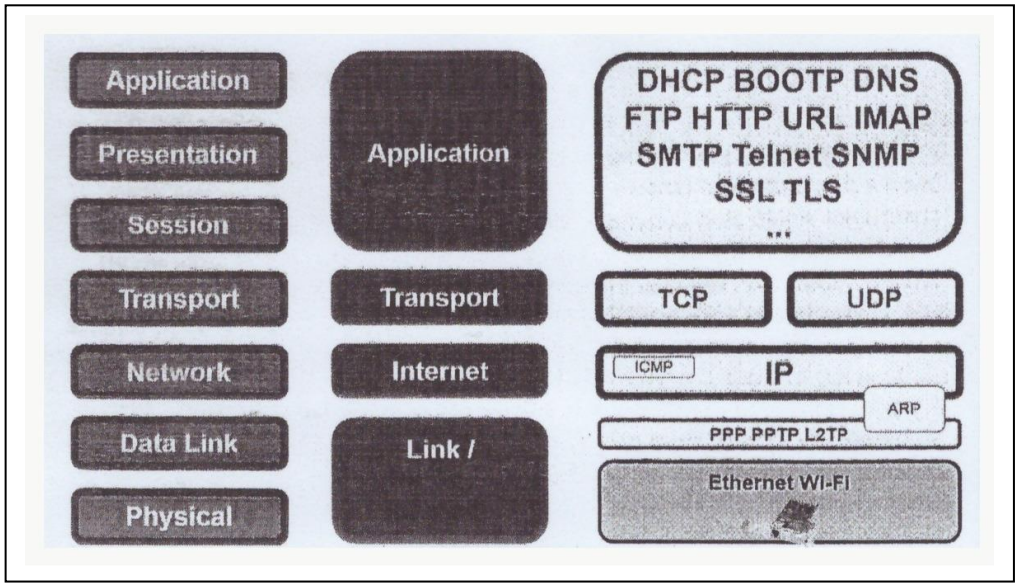

Figure 2.1 Basic architecture of OSI vs. TCP/IP

Source taken from Boston Technology Campus (2016). *CompTIA Network+ Certification (Exam N10-006) official study guide:* Study Notes

Using a layered structure as shown in Figure 2.1, the functionalities of a computer network is organised in a stack of layers. This is to facilitate design, flexible implementation, and intercommunication of computer network.

There is a virtual link between the corresponding layers in two communicating nodes. Data flow occurs in a vertical fashion: from the highest layer to the lowest

layer in a node, and then through the physical link to reach the lowest layer at the other node, and then following upwards to reach the highest layer in the stack. Each layer represents a well-defined and specific part of the system and provides certain services to the above layer. Accessible (by the upper layers) through interfaces, these services usually define what should be done in terms of network operations, but does not specifically define how such things are implemented.

The network simulation is no exception to the rule: it has to be able to incorporate the features of the OSI reference model. It also caters for future packages to be included and run transparently without disturbing existing components.

Based on the user demand at the application layer, a sending agent at the transport layer constructs packets and transmits them to a receiving agent at the corresponding transport layer through a low-level network.

2.3.1 Application layer

An application sits on top of the transport layer. It informs the attached agent (in transport layer) of user demand. The simulation environment used in this research classifies applications into traffic generators and simulated applications. A traffic generator creates user demand at a specific time as scheduled, while a simulated application acts all the way as if the application is running (Issariyakul and Hossain, 2008).

To make the research design as close as possible to the real world network, not only the CBR and FTP is used, but PackMimeHTTP as well. PackMineHTTP is an NS2 object that drives the generation of HTTP traffic. Each PackMimeHTTP object controls the operation of two types of applications, a PackMimeHTTP server application and a PackMimeHTTP client application. Each of these applications is connected to a TCP agent.

2.3.2 Transport layer

According to Figure 2.1, the network layer is responsible for delivering packets to any destination node within the network, and consequently, the routing of packets is one of the tasks of its protocols. IP is used by the two main protocols of the

transport layer, UDP and TCP, both providing packet flows between two hosts for application layer protocols.

UDP and TCP are among the most widely used transport layer protocols. There is a significant difference between these two protocols. While TCP provides a reliable transmission, the much simpler UDP only transmits packets and does not guarantee their delivery. In NS2, UDP is used for example by an application like CBR, while TCP is used by FTP and HTTP.

2.4 Network performance

The network can be defined as a collection of devices that inter-communicate, transmit, and receive data amongst themselves. An informal functional approach to the definition of network performance is "the speed of traffic through the network". Is speed everything? The answer is that speed is not everything. Also, the ability of a network to support large volumes of data transfer in the network, for instance, is part of the picture and as well some external clock source. Still, this is not yet everything for network performance. Network performance can be referred as a measure of service quality; it has been tested by different parameters. These parameters are called performance metrics and can be measured to evaluate the performance of a network. Table 2.1 displays a list of some metrics.

Table 2.1: Metrics list

Quantitative metrics	Qualitative metrics	Other aspects
Bandwidth	Good sound	CPU usage
Throughput	Bad picture	Memory usage
Latency	Slow response …	Chassis temperature
Jitter		Packet reordering …
Packet loss …		

Hasib (2006) argued that the most common network performance metrics are throughput and delay. Throughput (expressed in Bytes/second) can be defined as a measure of the data transfer per unit time between a sender and a receiver. Delay

(expressed in millisecond), on the other hand, is the time needed for a packet to travel.

These two metrics are the only considered network performance metrics of this research.

2.5 Review of previous researches done on VPN impact

Previous studies have been done to evaluate the performance and impact of VPN systems. The literature review reveals two aspects of the study that can be categorised in two, the impacting aspect and the impacted aspect.

2.5.1 The impacting aspect review

The focus of this review's aspect is on VPN factors affecting the performance of a network. The main factors of this aspect are: security services, algorithms, hardware, and software, just to name a few.

Extensive studies have been done on account of the impacting aspect.

Narayan *et al*. (2009) discussed the choice of VPN technologies and the associated algorithm that affects network performance. They proved that VPN protocols give different network performance metrics values for different combinations of operating systems, protocols and algorithms.

Kumar *et al*. (2002) presented a scenario consisting of various algorithms for provisioning VPNs. They concluded that even for the simple scenario in which network links are assumed to have infinite capacity, the VPN's algorithm's computation does have an impact on performance. They then proposed an improved model, but degradation still persisted.

Passito *et al*. (2005) carried out an analysis of IPSec overhead in 802.1 lb networks which showed a high overload in TCP and UDP traffic. The higher the security level, the higher the system overhead.

Lyu and Lau (2000) created a real-world environment and set up a secured firewall system. The objective was to study the performance of the firewall

system configured with different security policies and controls. Tests were designed for security verification and performance measurement.

Based on the test results; it was concluded that the firewall technology is a major factor for performance degradation. To put it another way, if frequent connections with small size data are required in communication, enhanced firewall security is likely to bring about performance degradation to the private network. However, their work disproved the existence of a security-performance relationship. The same level of security at firewall was found to be achievable with different firewall technologies. Inconsistent levels of network performance were noted as a result.

Pruthi (2009) presented a project and implementation of a WLAN with IPSec, an overview of security issues. He constructed an IPSec based VPN between the mobile node and the IPSec Gateway and investigated the performance of the UDP and TCP protocols. It was found that variation in packet loss rate and network throughput was affected.

Ashraf *et al*. (2009) were able to argue that even an unsecured VPN affects network performance; hence, the QoS experienced by the end-to-end users.

2.5.2 The impacted aspect review

On the other hand, this review's aspect focuses on the affected elements.

These elements are performance metrics such as bandwidth, throughput, packet loss, delay and so on.

Speaking about the impacted aspect, Powell (2010) showed that the use of a VPN for database transactions leads to a 446% delay of the query. Delays are also experienced in e-mail and FTP transactions.

Agarwal and Wang (2005) were able to demonstrate that throughput and delay are affected by different policies. They showed that authentication time may be a factor contributing towards QoS degradation than encryption overhead. This ascribed to hardware becoming faster.

Bourdoucen *et al*. (2009) simulated wireless LAN for IEEE802.11g protocol. The research group implemented a VPN by integrating an IPsec Virtual Private

Network technology. This was to secure the flow of traffic between the client and the server; OPNET WLAN utility was used for the purpose.

Their study underscored the importance of Response Time and Load of the Virtual Private Network over wireless LAN on performance.

Malik and Verma (2012) carried out simulation in the OPNET environment. In the simulation setup, voice and video applications were employed under two scenarios: without VPN and with VPN. Data relating to packet delay, traffic received and traffic sent was collected. The two researchers were able to demonstrate that the use of VPN increases the level of security; and in contrast, decreases network performance.

Rahimi *et al*. (2009) used J-Sim to simulate an MPLS for different packet drops, delays and throughputs. The study demonstrated a decrease in throughput and proportional losses in terms of packet losses.

2.6 Summarised findings

It is demonstrated that several studies have been done on the impact of VPN on network performance. The literature review performed concurs that performance is amongst the most important features in a network, regardless of the architecture. And different network architectures exist; LAN (Local Area Network), Metropolitan Area Network (MAN), Wide Area Network (WAN), VPN, Cloud, just to name a few. From all these architectures, VPN appears to take the lead.

Research on network performance in a VPN context has been previously considered. The key finding was that in most of the cases, there was a sensible variation over delay and network throughput. Throughput at the destination host was lower than that at the traffic source. And in terms of delays, traffic experienced proportional delays.

Besides, the VPN network security was a function of its algorithm.

The literature review also showed little research on the impacted factor (network performance). These factors may grow in importance in future. With powerful

simulations like NS2, this type of study is becoming possible. It is the intention of this research project to explore this and pave the way for further work.

Chapter 3: Research methodology, procedures and techniques

An account of a review of relevant literature was given in chapter 2 and subsequently, this chapter is devoted to a discussion of the design of the empirical investigation. Aspects that are discussed include the methodology of the research, the research instrument used, the data collection, the challenges and difficulties encountered. This chapter gives an overview of the research methodology adopted, the data collection technique and the simulation programme used.

3.1 Network simulations

Simulation can be defined as the recreation of real world scenarios using computer programmes. The technique enables one to perform extensive experimentation while relying on mathematical models of the system under investigation. The major advantage of simulations is that features of the model can be changed and the corresponding results are analysed cost-effectively in comparison to the real life experiments.

Network simulation tools can be categorised based on whether they are commercial or free. In the first instance, the network simulators do not provide the source code to the public for free; the users have to pay for the license. In the second instance, the open source network simulator has the advantage that the source code is open and free. The source code is also flexible and develops faster from the contribution of everyone involved. There is a wide variety of network simulators; Table 3.1 provides a list of the most common network simulators in the market.

Table 3.1: Network simulators list

Commercial	Open source
Matlab:(Physiomeproject,2015) Opnet: (Riverbed, 2012)	NS2: (Issariyakul and Hossain, 2008) Omnet++: (Omnet++, 2015) J-Sim: (Physiome project, 2015)

NS2 is arguably the most popular in academia because it is open-source, it has plenty of network and protocol objects library, and it is targeted at network research. From all the above, the research chooses NS2 as the research simulation. NS2 is a discrete event simulator where the advance of time depends on the timing of events which are maintained by a scheduler. NS2 is a combination of C++ language and OTCL script. The OTCL script is an object-oriented tool command line interpreter which interprets the scripts line by line the code written in notepad or in Microsoft-word etc. and save it by .TCL extension. Tool Command Language (TCL) is a language with a very simple syntax and allows a very easy integration with other languages. NS2 is installed on Linux and it has a command line interface.

Figure 3.1 shows the NS2 command line interface in Ubuntu Linux.

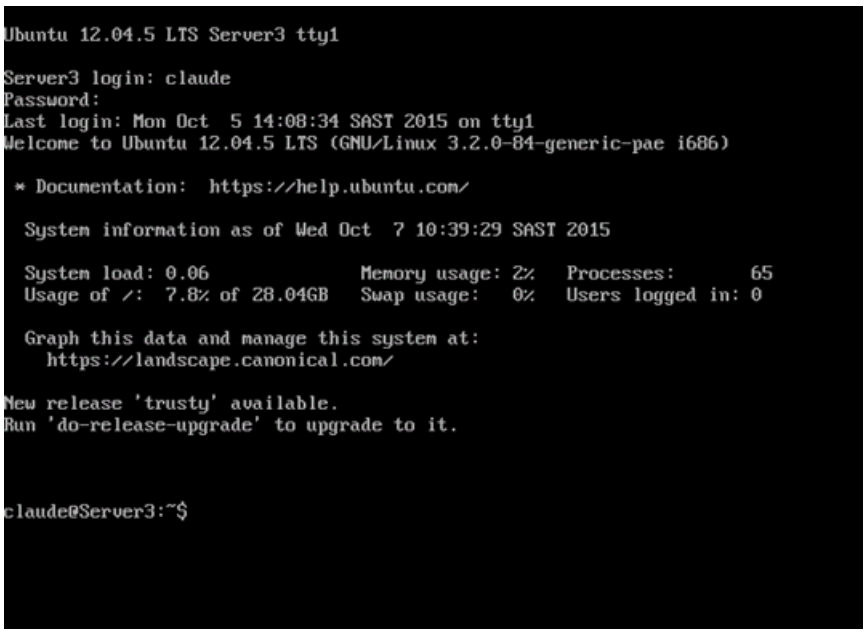

Figure 3.1: NS2 command line interface

The Linux command “Vim” (Figure 3.2) is used to create or open a file script in NS2. This can be achieved alternatively by opening a notepad file and save the file under the NS folder using a .tcl extension.

```
claude@Server3:~$ cd ns-allinone-2.35/
claude@Server3:~/ns-allinone-2.35$ vim scenario_1.tcl
```

Figure 3.2: Vim command to create or open tcl script

NS2 simulation starts with the command “set ns [new Simulator]” which is the first line in the TCL script to initialize the simulation. From initialisation, we define nodes, links, queue, topology, agents, and applications and trace files using varieties of command lines. The simulation can then begin using the command “$ns run”.

Figure 3.3 shows some of the command lines as it is inserted in a TCL script file.

```
set ns [new Simulator]
set mytracefilescen1 [open out_tracescen1.tr w]
set mynamtracefilescen1 [open out_namtracescen1.nam w]

$ns trace-all $mytracefilescen1
$ns namtrace-all $mynamtracefilescen1
#Closing the trace files
proc finish {} {
        global ns mytracefilescen1 mynamtracefilescen1
        $ns flush-trace
        close $mytracefilescen1
        close $mynamtracefilescen1
        #Call xgraph to display the results
        exec nam out_namtracescen1.nam &
        exit 0
}

# make nodes
set node1 [$ns node]
set node4 [$ns node]
# make links
$ns duplex-link $node1 $node4 100Mb 10ms DropTail

$ns run
```

Figure 3.3: Sample commands for TCL script

The NS2 output is either text-based or animation-based data. In order to analyse this data and study the network behaviour, the text-based data is formatted into a convenient form.

3.2 Simulation Setup

Considering that various factors contribute to network performance degradation, the test bed used a wired connection as opposed to the wireless one. Hence, some obvious initial degradation is cut off and makes the environment ideal. The simulation is deployed in two scenarios. Figure 3.4 presents the first scenario where the connection between the two nodes is a direct line, no VPN involved. This scenario consists of just 2 normal interconnected nodes, the research's baseline.

Figure 3.4: Scenario 1: Network without VPN

The second scenario (Figure 3.5), the VPN comes in the picture. This scenario consists of four interconnected nodes (2 hosts plus 2 extra nodes: routers with routing activated which form a VPN).

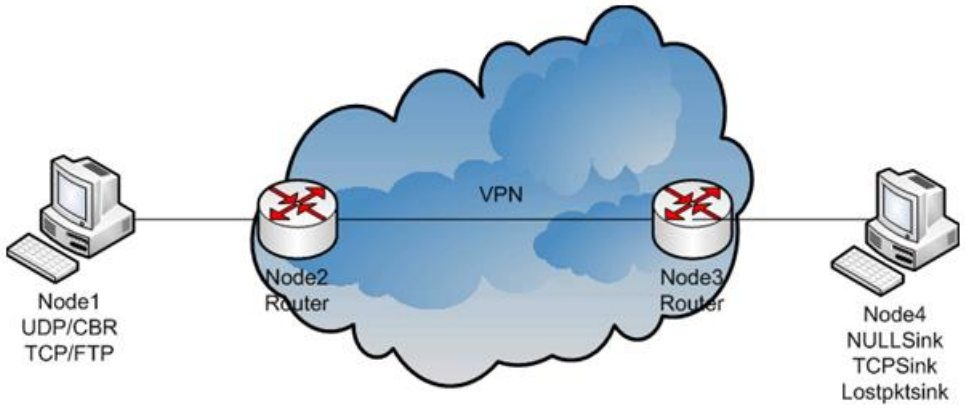

Figure 3.5: Scenario 2: Network with VPN

Figure 3.6 presents all processes and steps followed by the research from the implementation all the way to the findings. The writing of the simulation script, the running of the script, the data collection and its interpretation

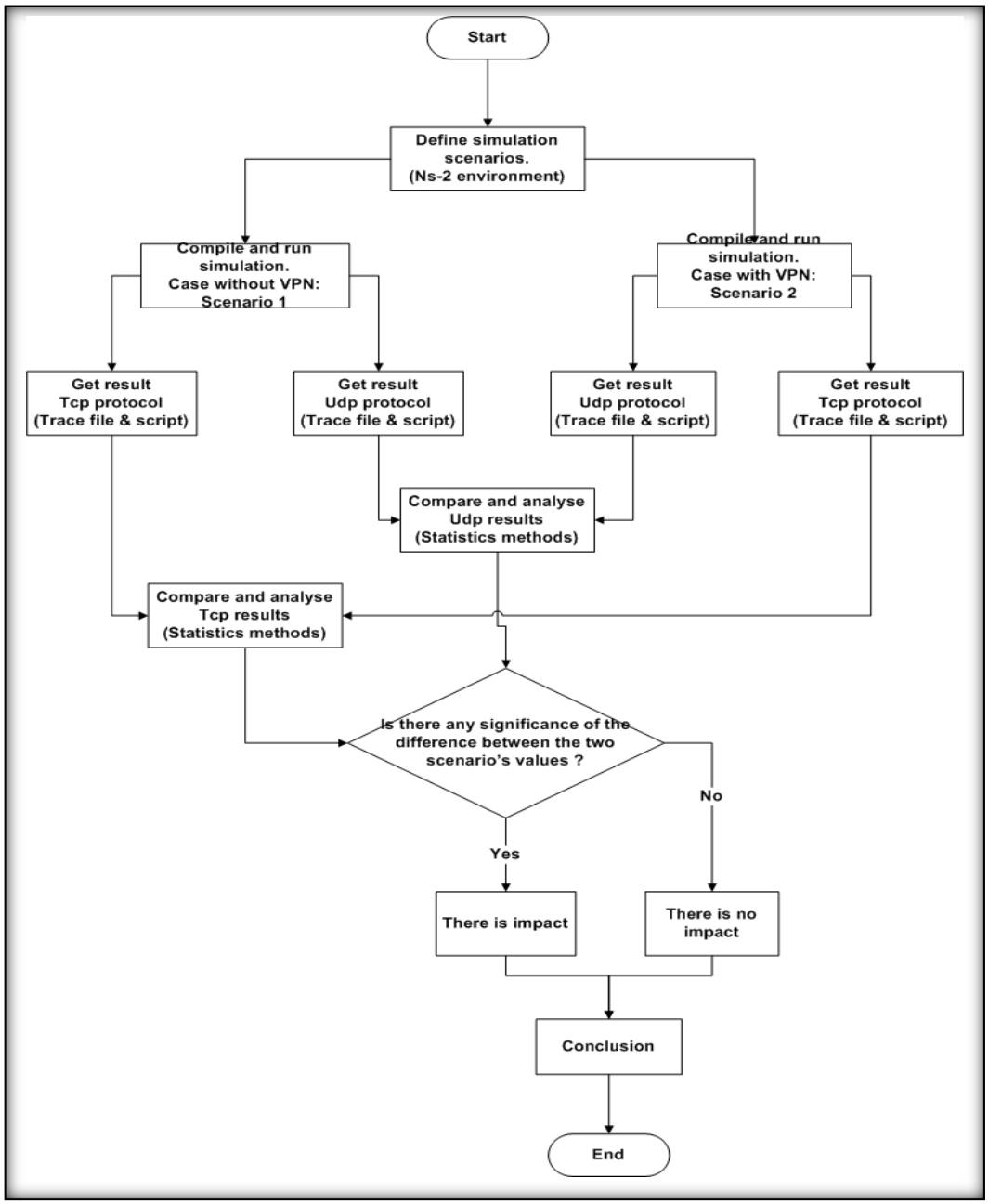

Figure 3.6: research flowchart

The simulation starts with the creation of the two scenarios, we define and setup scenario without VPN and scenario with VPN in within NS2. See index

Both scenarios are run using alternatively the two protocols which are TCP and UDP. Simulation's data are collected by protocol and by scenario using trace files. The collected data are then analysed and compared accordingly. If any significance of the difference is observed, then the impact existence is concluded. Otherwise no impact

To setup the test bed simulation, a CBR, FTP and HTTP applications are respectively generated following procedures in Figures 3.7 – 3.12.

3.3 Simulation Procedures

3.3.1 Procedure case scenario 1

For CBR application in Figure 3.7, the following procedures are implemented:

- Create node1 and node4:
  set node1 [ns node], set node4 [ns node]
- Create link between node1 and node4:
  $ns duplex-link $node1 $node4 100Mb 10ms DropTail
- Attach UDP and Sink to nodes
  $ns attach-agent $node1 $udp
  $ns attach-agent $node4 $lostpktsink_udp
- Attach CBR application to UDP agent
  $cbr attach-agent $udp
- Connect agent to sink
  $ns connect $udp $lostpktsink_udp

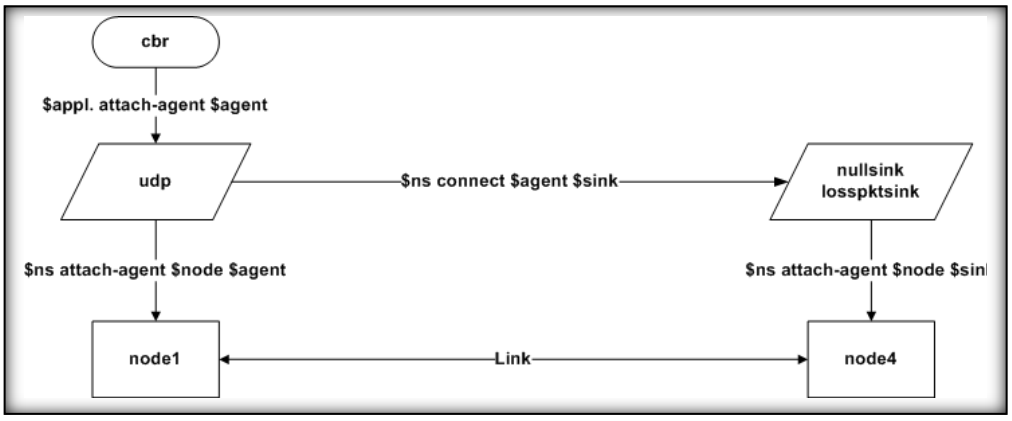

Figure 3.7: CBR application over UDP in scenario without VPN

In Figure 3.8, FTP application uses the same procedure as the previous.

The difference is the change in protocol and application itself; TCP and FTP as opposed respectively to the former, UDP and CBR.

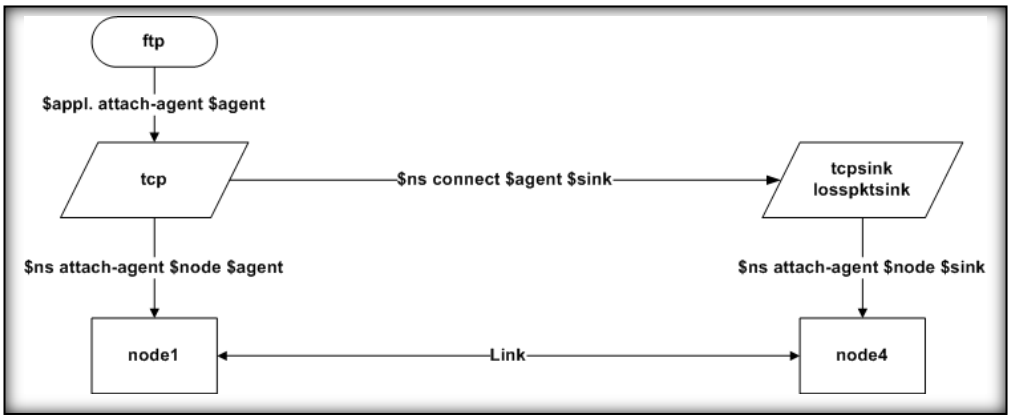

Figure 3.8: FTP application over TCP in scenario without VPN

In the HTTP implementation (Figure 3.9), beside the creation of nodes and link, the following procedure is applied:

- Create Client

  $pm set-client $node1
- Create Server

  $pm set-server $node4
- Create traffic

  $pm set-http-1.1

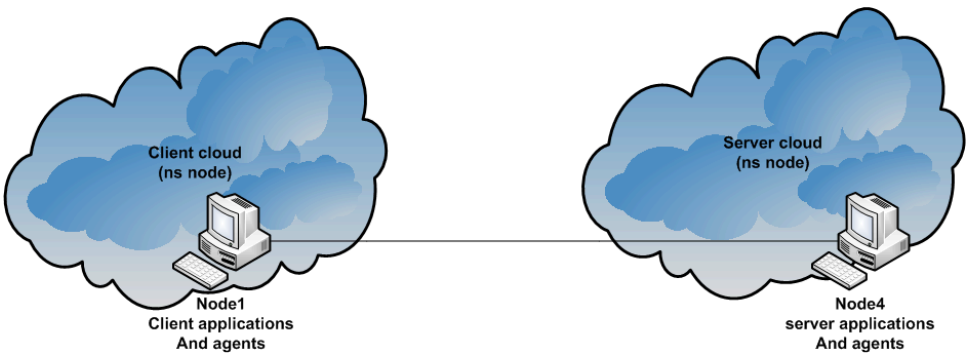

Figure 3.9: PackMimeHTTP Architecture in scenario without VPN

3.3.2 Procedure case scenario 2

Figure 3.10, same as scenario 1 above, plus two extra nodes in between. As well activate routing as follow:

- Create 2 extra nodes (node2 and node3)
- Create link between nodes

  $ns duplex-link $node1 $node2 100Mb 10ms DropTail

  $ns duplex-link $node2 $node3 100Mb 10ms DropTail

  $ns duplex-link $node3 $node4 100Mb 10ms DropTail
- Activate routing

  $ns rtproto DV

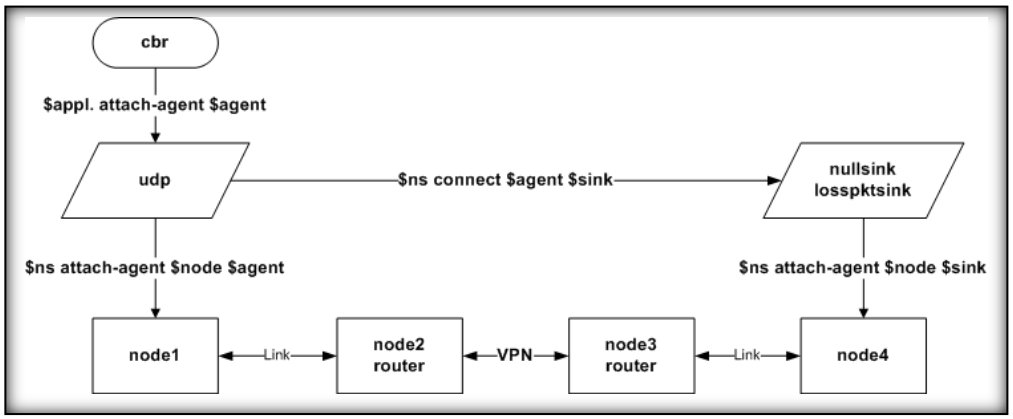

Figure 3.10: CBR application over UDP in scenario with VPN

The FTP uses same procedure than the above CBR except the change in protocol and application; TCP and FTP respectively (see Figure 3.11).

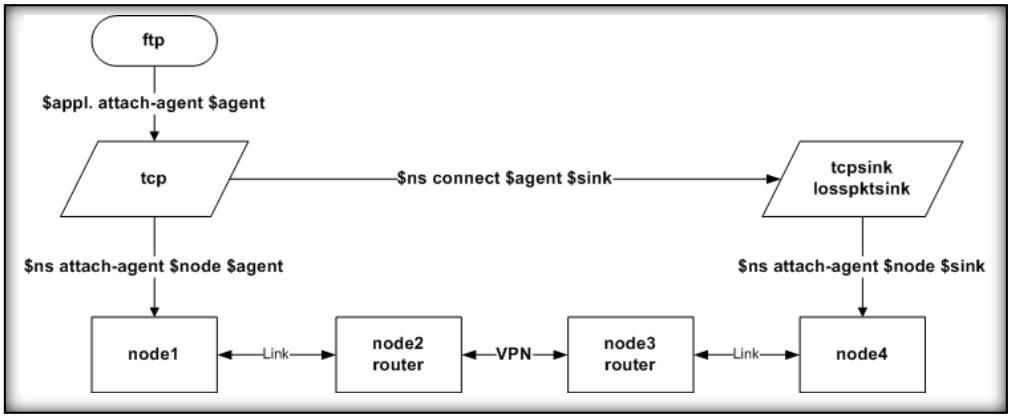

Figure 3.11: FTP application over TCP in scenario with VPN

Figure 3.12 HTTP, is same as scenario 1 HTTP except the two extra nodes and routing.

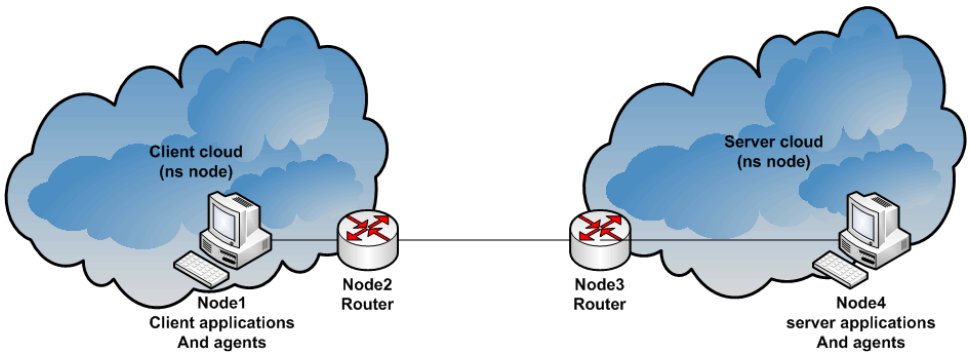

Figure 3.12: PackMimeHTTP Architecture in scenario with VPN

3.4 Simulation schedule and parameters

In this section, the scheduling of events is emphasised. Start and stop time of the simulation. As well as the parameters which trigger the results are introduced, packet size, window size, transfers rate and number of connections.

3.4.1 Schedule

The initializing command “set ns [new simulator]” creates an event scheduler, and events are then scheduled using the format:

$ns at <time> <event>

The scheduler starts through the command:

$ns run

In this simulation, the beginning and end of the three applications are set as follow:

$ns at 2 "$cbr start"

$ns at 7 "$cbr stop"

$ns at 8 "$ftp start"

$ns at 13 "$ftp stop"

$ns at 14 "$pm start"

$ns at 19 "$pm stop"

$ns at 20 "finish"

$ns run

Table 3.2 gives schedule resume for the three applications.

Table 3.2: Application schedule

Application	Start Time	Stop Time	Simulation
CBR	2 s	7 s	Models a user demand to periodically transmit data. Example: video and VoIP streams
FTP	8 s	13 s	Models a bulk data transfer
HTTP	14 s	19 s	Models a traffic generation on a link shared by many web clients and servers

3.4.2 Simulation parameters

For each scenario we ran the applications one by one, changing gradually the combination of settings (Tables 3.3, Tables 3.4 and Table 3.5)

Table 3.3: CBR parameters used for simulation

	Packet Size (Bytes)		
	512	**1024**	**2048**
CBR Transfer rate (Mb)			
10	X	X	X
100	X	X	X
1000	X	X	X

Table 3.4: FTP parameters used for simulation

	Packet Size (Bytes)		
	512	**1024**	**2048**
TCP window size (Mb)			
10	X	X	X
50	X	X	X
100	X	X	X

Table 3.5: HTTP parameters used for simulation

HTTP number of connection
5
10
15

3.5 Data collection

In this section, the recording and data collection method of the research is explained.

A Tcl script is compiled out of various NS2commands (See Appendices A, B, and C). This Tcl script is run, the trace result collected and stored in a text format file. Furthermore, the text format's trace is imported, processed and organised in Excel® format. The Excel® format is ready to be used for graphs and statistics such as means (averages). Trace file examples in an original format are illustrated in Figures 3.13 and 3.14

```
+ 2 0 1 cbr 512 ------- 0 0.0 1.0 0 0
- 2 0 1 cbr 512 ------- 0 0.0 1.0 0 0
r 2.010041 0 1 cbr 512 ------- 0 0.0 1.0 0 0
+ 2.5 0 1 cbr 512 ------- 0 0.0 1.0 1 1
- 2.5 0 1 cbr 512 ------- 0 0.0 1.0 1 1
r 2.510041 0 1 cbr 512 ------- 0 0.0 1.0 1 1
+ 3 0 1 cbr 512 ------- 0 0.0 1.0 2 2
- 3 0 1 cbr 512 ------- 0 0.0 1.0 2 2
r 3.010041 0 1 cbr 512 ------- 0 0.0 1.0 2 2
+ 3.5 0 1 cbr 512 ------- 0 0.0 1.0 3 3
- 3.5 0 1 cbr 512 ------- 0 0.0 1.0 3 3
```

Figure 3.13: Default NS2 trace file

```
2.5 0.00819199999999996
3 0.00819199999999996
3.5 0.00819199999999996
4 0.00819199999999996
4.5 0.00819199999999996
5 0.00819199999999996
5.5 0.00819199999999996
6 0.00819199999999996
6.5 0.00819199999999996
7 0.00819199999999996
```

Figure 3.14: Customised TCP throughput trace file

3.6 Statistical procedures and techniques

Data collected were analysed using a hypothesis t-test. A hypothesis test is a process by which an analyst tests a statistical hypothesis. The hypothesis to be tested is called the null hypothesis. The null hypothesis is rejected if the value of the test statistic falls within the critical region and is not rejected otherwise. The methodology used by the analyst depends on the nature of the data used, and the goals of the analysis. The goal as stipulated above is to either accept or reject the null hypothesis which is in our case the "no impact". In order to accept or reject this hypothesis, the research uses a t-test which is one of a number of hypothesis tests to depict statistical differences between two population means. Assuming that the variances are unequal, the research applies a two-tail test (inequality) of the t-test which states that:

If "(t Stat) < - (t Critical two-tail)"Or "(t Stat) > (t Critical two-tail)", we reject the null hypothesis. Otherwise; "- (t Critical two-tail) < (t Stat) < (t Critical two-tail)", the null hypothesis is accepted

To be noted:

- "(t Stat)" : t value calculated from the data
- "(t Critical two-tail)": the tabulated value; the value that we would need to exceed in order for the difference between the means to be significant at the 5% level (Ross, 2010; Neil, 2012)

3.7 Hardware specifications

A laptop of specifications as shown in Table 3.6 was used as simulation host.

Table 3.6: Hardware hosting simulation

Brand	Hewlett Packard
Model	Hp ProBook 4530
Processor name	Intel Core i3
Processor speeds	2.20 GHz
Ram	4.00 GB
Storage	500 GB
Ethernet	Realtek PCIe

3.8 Software specifications

Four software were used in this study as shown in Table 3.7

Table 3.7: Software used in the simulation

Software used	Role played
Microsoft Windows 7 Professional 32-bit	Host operating system
Oracle virtual box-4.3.12-93733-Win	Virtual machine running on top of windows
Ubuntu-14.04-server-i386	Linux version installed on the virtual machine
NS2.35 Simulator	Simulation software installed on Ubuntu

3.9 Challenges and difficulties encountered

The no access to a real life experimental system was the main limitation of the study. Besides, the use of a VM (Virtual Machine) created bugs which made the Nam (Network animator) unable to visualise real world live traffic.

Chapter 4: Effects of VPN on CBR application

4.1 Introduction

This chapter presents an analysis and evaluation of VPN's impact on CBR application. The research benchmarks Scenario 1 as the reference point that later on is referred back to compare against Scenario 2. This evaluation involves comparing the means of various active measurements (Calyam *et al.*, 2005) for throughput and delay respectively. Results are presented and analysed using graphs and hypothesis testing respectively

4.2 Collected results for CBR

Various tests are alternatively run using different aspects of transfer rates and packet size as per table 3.3 above. With CBR sitting on UDP protocol, the simulation results are summarised and presented in table 4.1 and table 4.2 below.

Table 4.1: Average CBR throughput

	Packet size					
	512 Bytes		**1024 Bytes**		**2048 Bytes**	
Transfer rate 10	A	B	A	B	A	B
	0.0073	0.0073	0.0145	0.0145	0.0291	0.0291
Transfer rate 100	A	B	A	B	A	B
	0.0073	0.0073	0.0145	0.0145	0.0291	0.0291
Transfer rate 1000	A	B	A	B	A	B
	0.0073	0.0073	0.0145	0.0145	0.0291	0.0291

Table 4.2: Average CBR delay

	Packet size					
	512 Bytes		1024 Bytes		2048 Bytes	
Transfer rate 10	A	B	A	B	A	B
	0.01004	0.01671	0.01008	0.02012	0.01013	0.02218
Transfer rate 100	A	B	A	B	A	B
	0.01004	0.01671	0.01008	0.02012	0.01013	0.02218
Transfer rate 1000	A	B	A	B	A	B
	0.01004	0.01671	0.01008	0.02012	0.01013	0.02218

4.4 Performance evaluation

This section evaluates the results obtained from the research simulation in section 4.3. Gathered CBR's throughput and delay data from the two different scenarios are presented and analysed.

4.4.1 Throughput

One of the metrics analysed in this section is throughput, it uses two parameters which are packet size and transfer rate.

It can be seen from Table 4.1 that column A equals column B for each packet size. And as the three transfer rates provide same insight, a random transfer rate and packet size is selected and its results presented on behalf of all. Figure 4.1 gives us an average CBR throughput graph of three different packet sizes: 512, 1024 and 2048 Bytes.

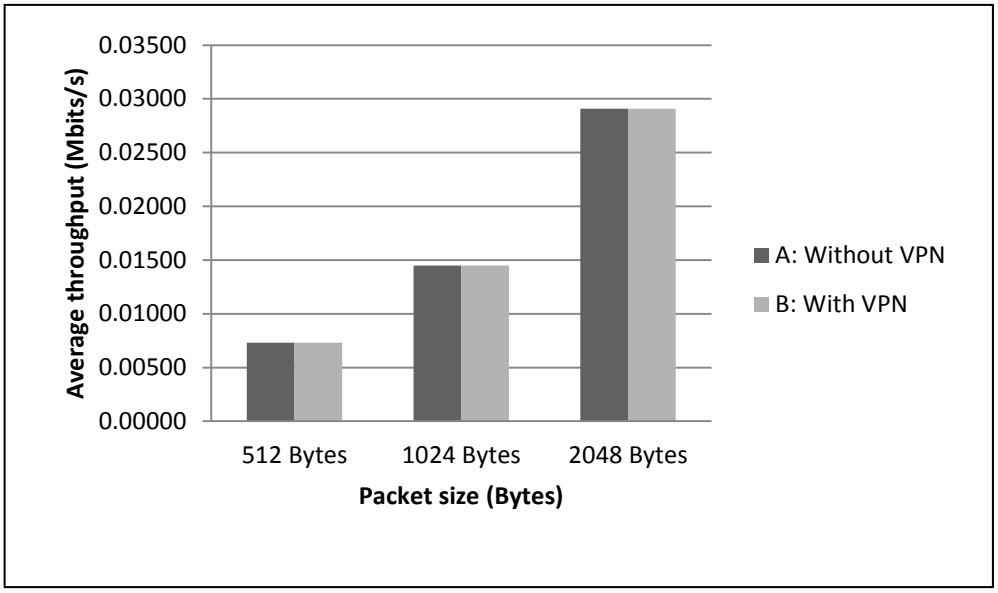

Figure 4.1: Average CBR throughput

From figure 4.1 it is clear that column A equals column B. No change, thus impact. Choosing randomly 1024 bytes' data from appendix E, the t-test gives the result in table 4.3

Table 4.3: Student's t-test for CBR throughput, packet size 1024

t-test: two-sample assuming unequal variances		
	With VPN	**Without VPN**
Mean	0.014545455	0.014545455
Variance	2.32727E-05	2.32727E-05
Observations	11	11
Hypothesized Mean Difference	0	
tSat	0	
t Critical two-tail	2.085963447	

From the above t-test table and applying the theory in section 3.6: "Statistical procedures and techniques", it is found that the condition

"- (t Critical two-tail) < (t Stat) < (t Critical two-tail)" is observed:

- 2.085963447 < 0 < 2.085963447. The null hypothesis is not rejected, no impact on this throughput.

4.4.2 Delay

The delay has a different approach than throughput. Column A differs from B in all the packet size cases, but the transfer rates present same insight for all 10, 100 and 1000 Mb. As observations for transfer rate provide the same insights, a random transfer rate of 1000 MB is selected and results presented. Figure 4.4 shows the average CBR delay graph of transfer rate 1000 Mb, packet size 512, 1024 and 2048 Bytes.

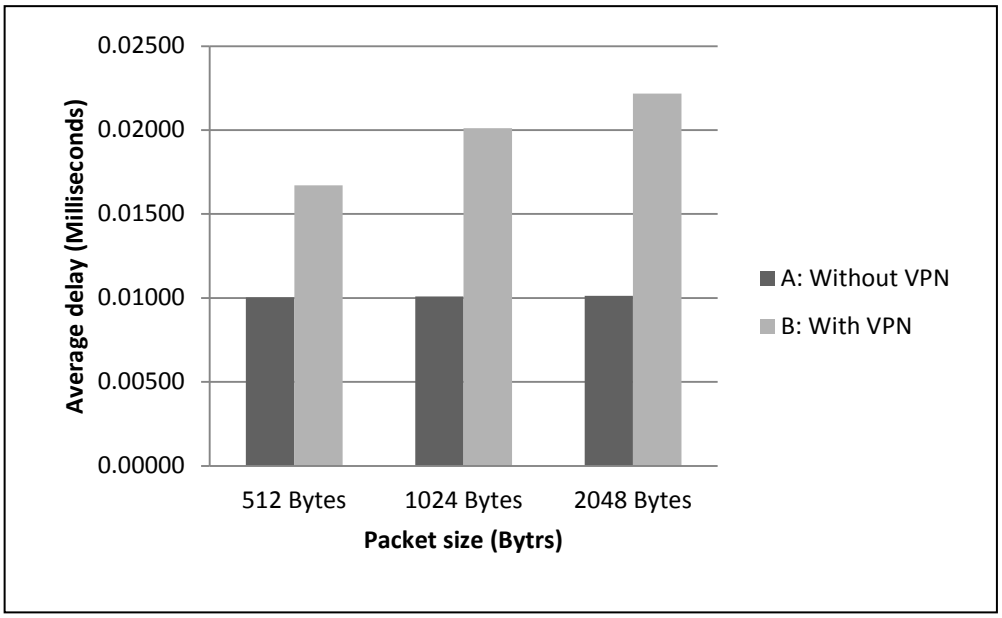

Figure 4.2: Average CBR delay of transfer rate 1000

From figure 4.2 above and among its three packet sizes, the 512 bytes is randomly selected and a t-test is computed out of it using its data from appendix G to give table 4.4

Table 4.4: Student's t-test for CBR delay, packet size 512

t-test: two-sample assuming unequal variances		
	With VPN	**Without VPN**
Mean	0.016707967	0.010041
Variance	9.30844E-05	3.34363E-36
Observations	30	10
Hypothesized Mean Difference	0	
tSat	-3.784866137	
t Critical two-tail	2.045229642	

Using the results from the above table in conjunction with the theory in section 3.6: "Statistical procedures and techniques", it is observed that

"(t Stat) < - (t Critical two-tail)" of the theory is proven;

-3.784866137 < -2.045229642; the null hypothesis is rejected.

These two scenarios differ, thus existence of impact on the delay.

Packet size 1024 bytes and 2048 bytes follow the same pattern than 512 bytes discussed above. The student's t-test in Table 4.5 combines both two results

Table 4.5: Student's t-test for CBR delay, packet size 1024 And 2048

t-test: two-sample assuming unequal variances				
	Packet size 1024		**Packet size 2048**	
	With VPN	**Without VPN**	**With VPN**	**Without VPN**
Mean	0.02012	0.01008	0.02218	0.01013
Variance	0.00011	1.05E-12	0.0001	1.55E-09
Observations	40	20	50	30
Hypothesized Mean Difference	0		0	
tSat	-6.19529		-8.47857	
t Critical two-tail	2.02269		2.00958	

The results in Table 4.5 Show that the null hypothesis can be rejected. In both cases the following condition "(t Stat) < - (t Critical two-tail)" is observed:
See comments in table 4.6

Table 4.6: Comments and explanation

Packet size 1024	Packet size 2048
(t Stat) < - (t Critical two-tail). -6.19529 < -2.02269 The null hypothesis is rejected Thus impact	"(t Stat) < - (t Critical two-tail)" -8.47857< -2.00958 The null hypothesis is rejected Thus impact

In both Cases, the delay is affected

4.5 Summarised findings

Concerning throughput, all traffic injected was received regardless of packet size or window size. It is proved that there is nothing lost in the UDP's network journey; consequently, throughput A and throughput B remain constant. No impact observed for throughput.

On the delay account, the student's test proved that the average delay in A scenarios was not the same compared to the B scenarios. Thus, the delay was influenced

Chapter 5: Effect of VPN on FTP application

5.1 Introduction

In this chapter, results of the investigation in line with the FTP application is analysed, interpreted and presented. The same statistical tool, t-test of section 3.6 "Statistical procedures and techniques" is used to test the no impact hypothesis.

5.2 Simulation results for FTP

Tests are run using different aspects of window sizes and packet sizes as shown in Table 3.4. With FTP using TCP protocol, the experimental results are summarised and presented in Table 5.1 and Table 5.2 below.

Table 5.1: Average FTP throughput

	Packet size					
	512 Bytes		**1024 Bytes**		**2048 Bytes**	
Window size 10	A	B	A	B	A	B
	1.7857	0.6464	3.8164	1.2427	0.0291	0.0291
Window size 50	A	B	A	B	A	B
	9.7146	3.1382	18.9	6.0546	36.9309	11.8773
Window size 100	A	B	A	B	A	B
	19.5318	6.2052	37.6418	11.9573	73.4173	23.31

Table 5.2: Average FTP delay

	Packet size					
	512 Bytes		1024 Bytes		2048 Bytes	
Window size 10	A	B	A	B	A	B
	0.01	0.03	0.01	0.03	0.01	0.03
Window size 50	A	B	A	B	A	B
	0.01	0.03	0.01	0.03	0.01	0.03
Window size 100	A	B	A	B	A	B
	0.01	0.03	0.01	0.03	0.01	0.03

5.4 Throughput's performance evaluation

Window size is used to trigger results in turns with 10Kb, 50Kb and 100Kb

5.4.1 Window size 10 Kb

Figure 5.1 shows the average TCP throughput graph of window size 10 Kb, packet size 512, 1024 and 2048 Bytes.

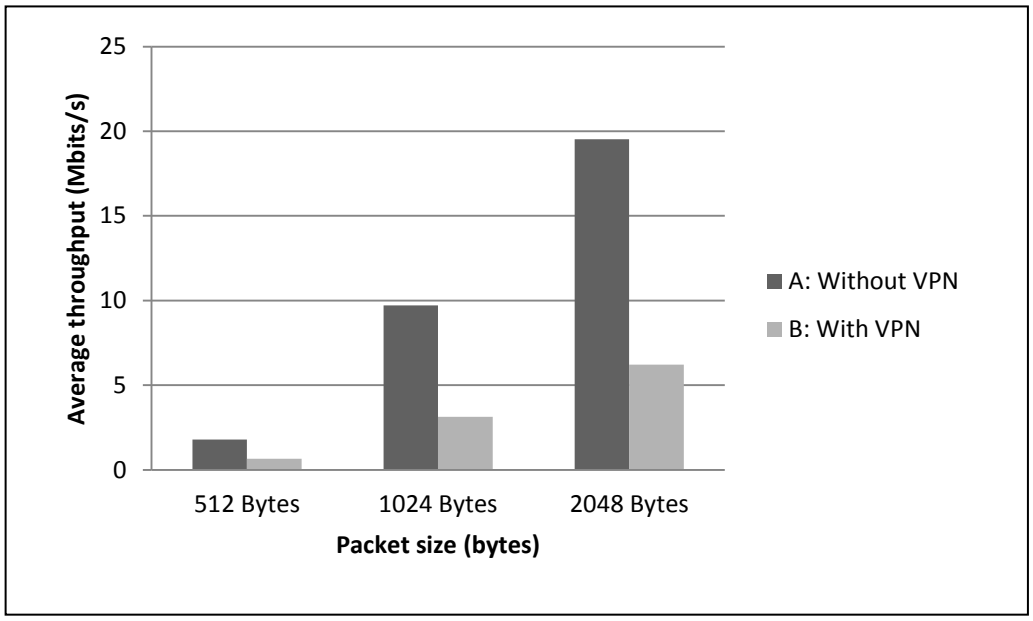

Figure 5.1: Average TCP throughput of window size 10

To conclude this case, the t-test is used for each of the three different packet sizes. For these different situations, table 5.3 and table 5.4 are respectively presented as follow:

Table 5.3: t-test for FTP throughput, packet size 512

t-test: two-sample assuming unequal variances		
	With VPN	**Without VPN**
Mean	0.646363636	1.785727273
Variance	0.053165455	0.746839218
Observations	11	11
Hypothesised Mean Difference	0	
tSat	4.2248611	
t Critical two-tail	2.20098516	

Referring to the statistical tool of section 3.6, the below condition of hypothesis rejection is met:

"(t Stat) > (t Critical two-tail)" which from the table gives

4.2248611 > 2.20098516

This implies the existence of a difference in Scenarios A and B. There is an impact on this throughput case.

Follows is the window size test of still 10 Kb but the packet size is changed from 512 to 1024 and 2048 Bytes. The results of a combined t-test are shown in Table 5.4.

Table 5.4: T-test for FTP throughput, packet size 1024 and 2048

t-test: two-sample assuming unequal variances				
	Packet size 1024		Packet size 2048	
	With VPN	Without VPN	With VPN	Without VPN
Mean	1.242727273	3.816363636	7.4518182	2.4409091
Variance	0.197861818	1.621405455	6.1771164	0.7655291
Observations	11	11	11	11
Hypothesized Mean Difference	0		0	
tSat	6.328419693		6.3074	
t Critical two-tail	2.17881283		2.1788128	

From table 5.4 the condition of "(t Stat) > (t Critical two-tail)" is met for both 1024 bytes and 2048 bytes as displayed in table 5.5

Table 5.5: Finding's comments for Window size 10, packet size 1024 and 2048

Packet size 1024	Packet size 2048
(t Stat) > (t Critical two-tail) 6.328419693 > 2.17881283 The null hypothesis is rejected Thus impact	"(t Stat) > (t Critical two-tail)" 6.3074 > 2.1788128 The null hypothesis is rejected Thus impact

The null hypothesis does not stand. The observation done is that the two scenarios average throughputs are different in both cases. This indicates that throughput is affected. This is enough evidence to agree on the inequality of the average throughput between scenarios. There is an impact on these throughput cases.

5.4.2 Window size 50

Below (Figure 5.2) is a pictorial comparison between the average scenario TCP throughput of window size 50 Kb, packet size 512, 1024 and 2048 Bytes.

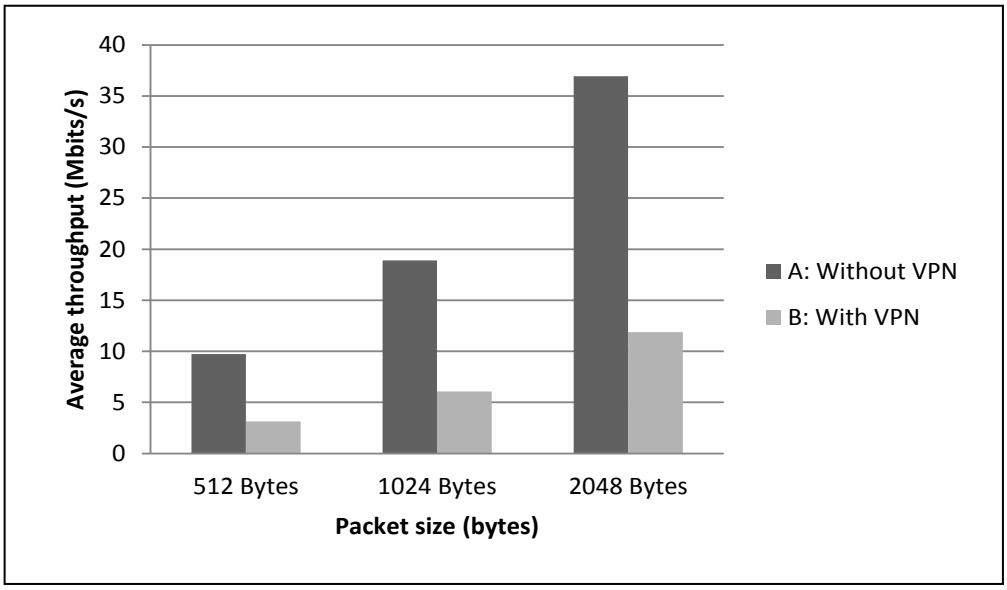

Figure 5.2: Average TCP throughput of window size 50

Not to be repetitive, we present straight the findings. These findings verify the " (t Stat) > (t Critical two-tail)" condition as proven in table 5.6

Table 5.6: Finding's comments for Window size 50

Packet size 512	"(t Stat) > (t Critical two-tail)" 6.200745254 > 2.160368656 The null hypothesis is rejected Thus impact
Packet size 1024	"(t Stat) > (t Critical two-tail)" 6.24175476 > 2.160368656 The null hypothesis is rejected Thus impact
Packet size 2048	"(t Stat) > (t Critical two-tail)" 6.2213516 > 2.1603687 The null hypothesis is rejected Thus impact

5.4.3 Window size 100

For the next window size which is 100 Kb, we present it visual representation as shown in figure 5.3 and give it summary interpretation as per the t-test null hypothesis.

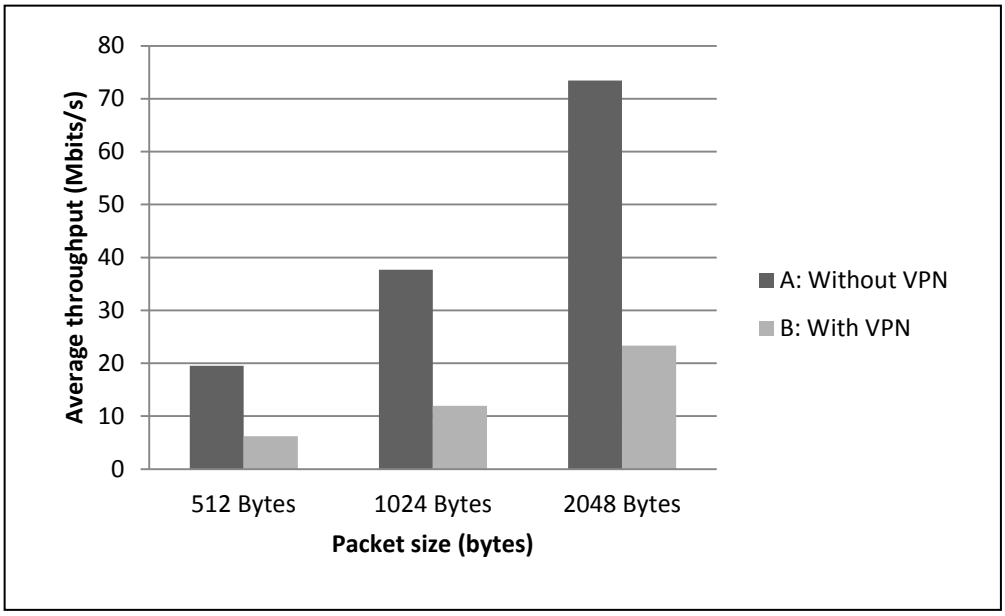

Figure 5.3: Average TCP throughput of window size 100

Examining this graph, it is found that the case follows the same pattern as the previous two and the "(t Stat) > (t Critical two-tail)" condition is met for all the three packet sizes

Table 5.7: Finding's comments for Window size 100

Packet size 512	6.179818204 > 2.160368656 The null hypothesis is rejected. Impact
Packet size 1024	6.177265158 > 2.160368656 The null hypothesis is rejected. Impact
Packet size 2048	6.1577338 > 2.1603687 The null hypothesis is rejected. Impact

5.5 Delay's performance evaluation

Looking at data from figure 5.2, it is noticed that regardless of the window size and packet size, we have the same pattern for all. Due to space limitation and not to be exhaustive, we present and discuss the only one set of case on behalf of all Figure 5.4 shows this average FTP delay graphically.

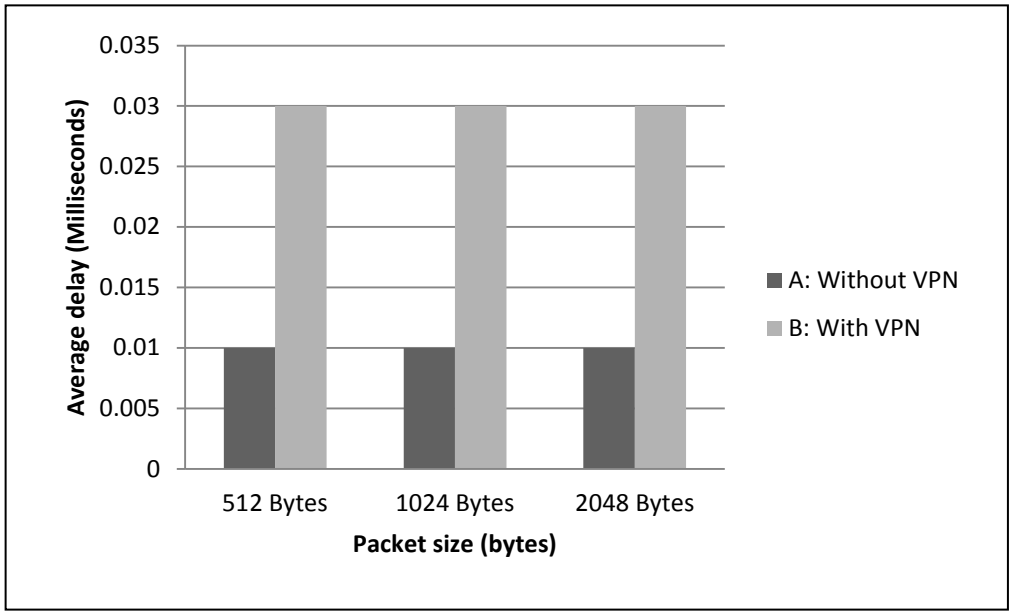

Figure 5.4: Average FTP delay

In all the cases, it is observed that all the A columns are clearly smaller than the B columns.

5.6 Summarised findings

From the above, this is enough evidence to imply on the inequality between the two scenarios. Throughputs and delay are impacted in all cases.

Chapter 6: Effects of VPN on HTTP application

6.1 Introduction

This chapter analysis and evaluates the VPN's impact on HTTP application using PackMimeHTTP. Gathered results are presented and commented respectively using graphs and hypothesis t-test as per section 3.6: "Statistical procedures and techniques".

6.2 Simulation results for HTTP

Simulations are run off different connection rates on the HTTP throughput and delay of the network. Averages of the two scenarios are compared on the account of throughput and delay. Connection rates of 5, 10, and 15 are in turns the triggers parameters that are used in this simulation.

Gathered experimental results are summarised and presented in table 6.1 and 6.2.

Table 6.1: Average HTTP throughput

	A: Without VPN	**B: With VPN**
Connection rate 5	49.49358225	17.139651
Connection rate 10	39.46256811	16.583419
Connection rate 15	44.45390085	16.170693

Table 6.2: Average HTTP delay

	A: Without VPN	**B: With VPN**
Connection rate 5	0.01	0.03
Connection rate 10	0.01	0.03
Connection rate 15	0.01	0.03

6.4 Performance evaluation

This evaluation is presented for the account of throughput and delay. Our protocol of interest in this specific case is the TCP which is the HTTP application's backbone.

6.4.1 Throughput

The average HTTP throughput graph of connection rate 5, 10 and 15 is presented in figure 6.1.

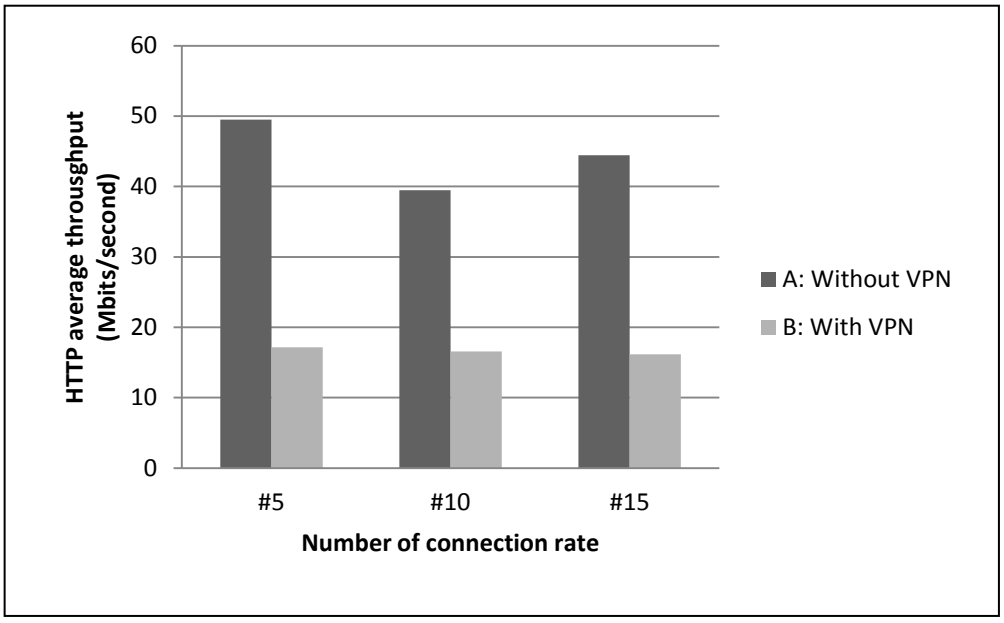

Figure 6.1: Average Http throughput

From the above graphical representation, it is found that the "(t Stat) > (t Critical two-tail)" condition is met for all the three connection rate numbers as per the table below.

Table 6.3: HTTP finding comments

Connection rate 5	3.132110036 > 2.004044783 The null hypothesis is rejected. Impact
Connection rate 10	3.955371321 > 1.980992298 The null hypothesis is rejected. Impact
Connection rate 15	5.572167228 > 1.974715786 The null hypothesis is rejected. Impact

To emphasise this difference between scenario A and scenario B, the instant throughput versus time graph is introduced. When comparing these two scenarios instant throughput, the shapes change their curves

Figure 6.2, figure 6.3, and figure 6.4 reflect these changes.

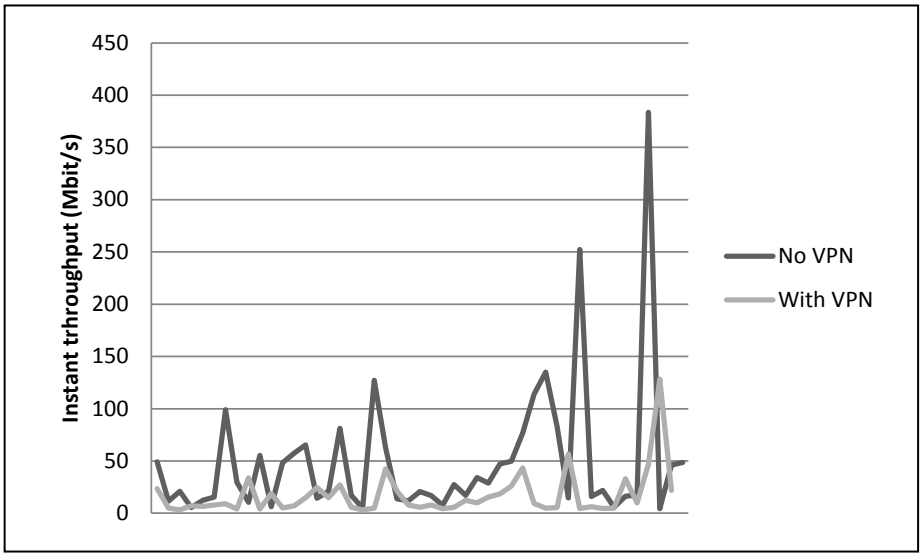

Figure 6.2: Comparison of instant throughput, connection rate 5

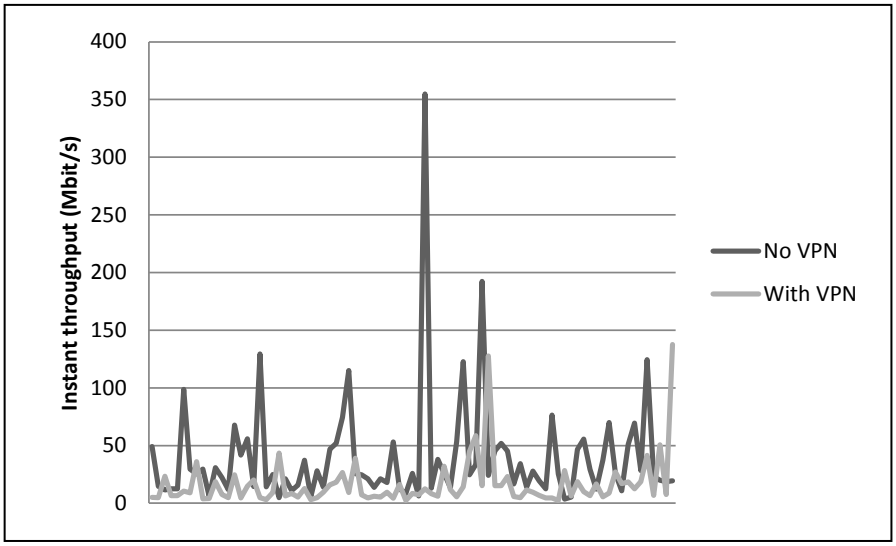

Figure 6.3: Comparison of instant throughput, connection rate 10

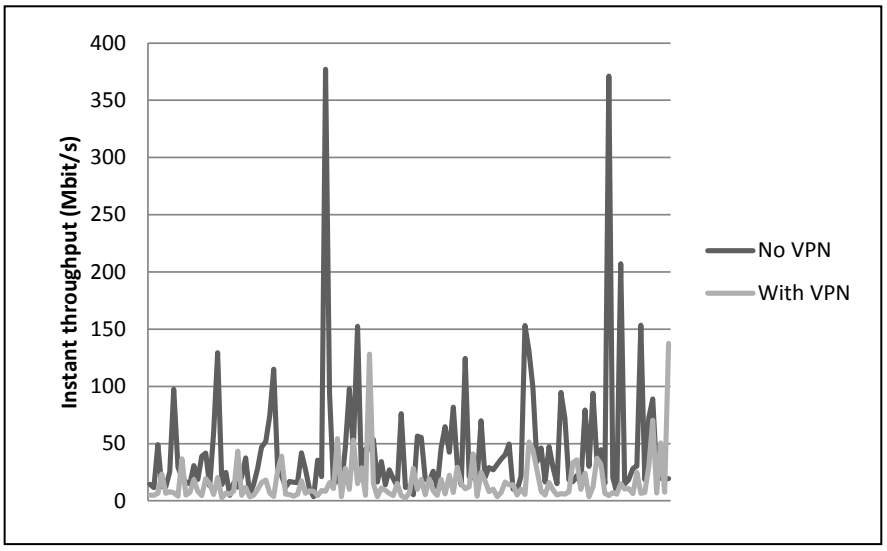

Figure 6.4: Comparison of instant throughput, connection rate 15

From the instant throughput perspective, it is observed that the effect of VPN brings a decrease in the link utilisation and capacity. The results of this process are viewable in the above figures and tables.

There is a difference, thus impact but no linear relationship exists between scenario A and scenario B.

6.4.2 Delay

HTTP spends more time waiting than it does transfer data.

For this raison, the HTTP delay is an important metric to look after.

Figure 6.5 shows an average HTTP delay graph of connection rate 5, 10 and 15.

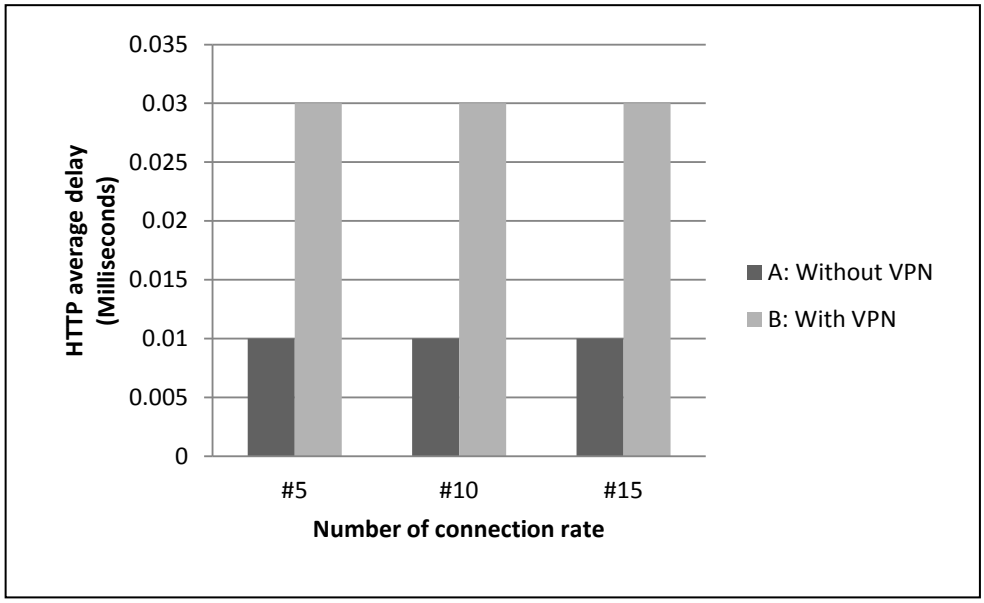

Figure 6.5: Average Http delay

Table 6.4 gives the computed results which are the same for all the three number of connections.

Table 6.4: T-test for HTTP delay, number of connection 5

t-test: two-sample assuming unequal variances		
	With VPN	**Without VPN**
Mean	0.029840157	0.010050758
Variance	5.0398E-06	1.01529E-08
Observations	1350	1345
Hypothesized Mean Difference	0	
tSat	-323.5595961	
t Critical two-tail	1.961717572	

The t-test condition meeting the requirement of the no rejection of the null hypothesis as stipulated in section 3.6 is:

“(t Stat) < - (t Critical two-tail)”

-323.5595961 < -1.961717572; the null hypothesis is rejected.

These two scenarios differ significantly, thus the existence of impact on the delay.

6.5 Summarised findings

Both HTTP throughput and delay are impacted by the VPN, but these impacts at the VPN level are not proportional to the number of connection.

Chapter 7: Conclusion and recommendations

7.1 Introduction

VPN plays an important role in the industry in the sense that it helps keep productivity life. From where ever, one can always connect back to the office and uses resources. However, it is understood that in terms of performance, a VPN-based system is relatively slow when compares with a LAN direct connection. This research commuted to determine the impact of VPN usage on network performance.

7.2 Summary and conclusion

The goal of this research was to study network performance impact due to the use of VPNs. To this end, simulations were set in the NS2 environment, a software simulator for studying network systems. Relevant data output was generated and collected under various testing conditions. Results were analysed and compared in terms of throughput and delay averages using Student's t-test. For the different simulation conditions, a 100 Mbps was considered. In all the cases, the two scenarios were simulated: network without VPN and network with VPN.

The main specific objectives were 1) to identify common network performance metrics and evaluate its VPN's impact, 2) compare the VPN's impact from different common network applications perspective and 3) to recommend on how to reduce this VPN network performance impact.

Therefore, as to respond to the research questions the following conclusions can be said:

Firstly, the research produced a number of findings that have helped to generate a deeper understanding of how much the network performances are impacted by the VPN.

For the first objective, the results revealed that the common network performance metrics are throughput and delay. In terms of throughput, the UDP throughput

was found to be unaffected by the VPN regardless of the CBR transfer rates and packet sizes tested. However, it was learned that the TCP throughput, as opposed to the UDP, was influenced by the VPN. This influence was a function of both packet size and window size.

In terms of the delay, it was learned that the UDP delay was sensitive to the VPN. Packet size was one of the influential parameters for this impact. Moreover, it was discovered that the transfer rate does not influence the UDP delay.

The TCP delay was found to be sensitive to the VPN and packet size was the only one influential parameter for this sensitivity. Window size did not influence the TCP delay.

As for the HTTP, it was learned that the HTTP throughput and HTTP delay was affected by the VPN. The impact was related to the Connection rate but no linear relationship exists between scenarios.

In achieving objective 2, we had a clear view and understanding that the impact of the VPN depends on the protocol used. UDP is a datagram packet which has less overhead. TCP, however, is a connection-based protocol which assumes an unreliable medium. Consequently, the TCP has more overhead and encounters adverse effects. Results proved in general that TCP throughput as opposed to UDP throughput measured in scenario A is very much lower compared to the TCP throughput measured in scenario B. Thus the FTP throughput and HTTP throughput which are applications sitting on TCP protocol are impacted but the CBR application sitting on UDP is not. Hence CBR, FTP and HTTP are differently influenced.

For objective 3, the increase in delay is justified by the decrease in throughput and vice versa. These two conditions can be accordingly rectified or improved by choosing proper hardware, proper software, well - selected encryption algorithms at the VPN level, and setting up configuration parameters of likes of small window size, small packet size and small connection rate.

Although it can be improved by using faster hardware and better algorithms, throughput is a critical performance metric which will limit the number of users whom the VPN can support. Thus, higher throughput is better.

The more you decrease the packet size and window size, the less is the impact.

7.3 Recommendations and future work

The design should be extended so as to include other factors that are known to potentially affect performance. For this, an extension of the simulation script is required. The use of other simulation packages should be explored in addition to NS2.

In the case of a simulation is used, this should be run on a physical machine rather than a VM.

Finally, a validation programme whereby live experiments are carried out is necessary to complete the work.

References

Agarwal, A.K., Wang, W. 2005'Measuring performance impact of security protocols in wireless local area networks', *2nd International Conference on Broadband Networks*, 2005. North Carolina: IEEE

Ashraf, M.T. Davies, J.N Grout, V. 2009, *An investigation into the effect of security on performance in a VoIP network*, Glyndwr University Research Online, viewed 27 August 2014, <www.glyndwr.ac.uk/computing/research/pubs/sein_adg.pdf >

Bourdoucen, H., Al Naamany, A., & Al Kalbani, A. 2009, 'Impact of implementing VPN to secure Wireless LAN'. *International Journal of Computer, Information and Systems Science*, vol. 3, no. 1, pp. 17

Calyam, P., Krymskiy, D., Sridharan, M., & Schopis, P., 2005 'Active and passive measurements on campus, regional and national network backbone paths'. *In 14th International Conference Computer, Communications and Networks*. ICCCN 2005. USA: IEEE

Boston Technology Campus 2016, *ComTIA Network+ Certification Official study guide: Study Notes*, g525eng ver075. Gtslearning

Fisli, R. 2005, '*Secure corporate communications over VPN-based WANs*', Master's degree', Royal Institute of Technology

Hasib, M. 2006, '*Analysis of packet loss probing in packet networks', PhD thesis*, University of London

Issariyakul, T. & Hossain, E. 2008, *Introduction to network simulator NS2*, 2nd edition. Springer, New York

Kothari, C.R. 2004, *Research methodology, methods and techniques*, 2nd revised edition, New Age International (Pty) Ltd, New Delhi

Kumar, A. Rastogi, R. & Silberschatz, A. 2002 'Algorithms for provisioning virtual private petworks in the hose model'. *In ACM Transactions on networking*. United States: IEEE

Lyu, M.R. & Lau, K.Y. 2000 'Firewall security: Policies, testing and performance evaluation'. *In the 24th Annual International Computer Software and Applications Conference*, COMPSAC 2000. Taipei: IEEE

Malik, A. & Verma, H. 2012, 'Performance analysis of virtual private network for securing voice and video traffic', *International Journal of Computer Applications*, vol. 46, no. 16, pp. 25 – 29

Mathworks 2015, *Mathworks*. *Matlab academy*. Viewed on 6 May 2015, <http://www.mathworks.com>.

Narayan, S., Brooking, K., & de Vere, S. 2009, 'Network performance analysis of VPN protocols: An empirical comparison on different operating systems'. *In International Conference on Networks Security, Wireless Communications and Trusted Computing*, Auckland: IEEE

Neil, A.W. 2012, *Introductory statistics*, 9th Edition. Addison-Wesley, Arizona State University: United States:

Omnet++ 2015, *Discrete event simulator, Omnet.* Viewed on 5 August, 2015 <https://omnetpp.org>.

Passito, A., Mota, E., Aguiar, R., Carvalho, L.S.G., Moura, E., Nascimento, E., Biris, I., Mota, E. 2005, 'Evaluating voice speech quality in 802.11b networks with VPN / IPsec'. *In Proceedings of the 13th IEEE International Conference on Networks*, Kuala Lumpur: IEEE

Physiome project 2015, *JSim, Physiome*. Viewed on 7 September, 2015 <http://www.physiome.org/jsim>

Powell, J.M. 2010, '*The impact of virtual private network (VPN) on a company's network*', Research report for Honour's degree, Utah State University

Pruthi, S. & Pruthi, G. 2009, 'Performance evaluation of wireless IPsec VPN'. *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 1, pp. 41 – 44

Rahimi, M.R., Hashim, H. & Rahman, R. 2009, 'implementation of quality of service (QoS) in multi-protocol label switching (MPLS) networks'. *In the 5th*

International Colloquium on Signal Processing and its Applications, CSPA. Kuala Lumpur: IEEE

Riverbed, 2012, *Opnet, Riverbed.* Viewed on 9 September, 2015 <http://za.riverbed.com/products/performance-managementcontrol/opnet.html?redirect=opnet>

Ross, SM. 2010, I*ntroductory Statistics*, 3rd Edition. British Library, Canada

Tan, NK. 2003, *Building VPNs with IPsec and MPLS*, New York, McGraw-Hill networking, USA

Appendices

Appendix A: Tcl script for scenario 1

```
##########TCL SCRIPT#####################################
set ns [new Simulator]
set mytracefilescen1 [open out_tracescen1.tr w]
set mynamtracefilescen1 [open out_namtracescen1.nam w]
set frputfile_udp [open out_frputscen1_udp.tr w]
set frputfile_tcp [open out_frputscen1_tcp.tr w]
set frputpktlost_udp [open out_lostpktscen1_udp.tr w]
set frputpktlost_tcp [open out_lostpktscen1_tcp.tr w]
$ns trace-all $mytracefilescen1
$ns namtrace-all $mynamtracefilescen1
#Closing the trace files
proc finish {} {
        global ns frputpktlost_udp frputpktlost_tcp frputfile_udp frputfile_tcp
mytracefilescen1 mynamtracefilescen1
        $ns flush-trace
        close $frputpktlost_udp
        close $frputpktlost_tcp
        close $frputfile_udp
        close $frputfile_tcp
        close $mytracefilescen1
        close $mynamtracefilescen1
        #Call xgraph to display the results
        #exec nam out_namtracescen1.nam &

        #exec xgraph out_frputscen1_udp.tr -t "Udp Throughput" &
        #exec xgraph out_frputscen1_tcp.tr -t "Tcp Throughput" &
        #exec xgraph out_lostpktscen1_udp.tr -t "Udp Lost" &

        exec awk -f delay.awk out_tracescen1.tr > delayfilescen1.dat
```

```
        #exec xgraph out_lostpktscen1_tcp.tr -t "Tcp Lost" &
        exit 0
}

#
# make nodes
set node1 [$ns node]
set node4 [$ns node]

# make links
$ns duplex-link $node1 $node4 100Mb 10ms DropTail

# Creation of sending agent
set udp [new Agent/UDP]
set tcp [new Agent/TCP]

#Creation of receiving agent / Sink
set nullsink [new Agent/Null]
set tcpsink [new Agent/TCPSink]
set lostpktsink_udp [new Agent/LossMonitor]
set lostpktsink_tcp [new Agent/LossMonitor]


# Creation of application (This is a modelling of users demand):
set cbr [new Application/Traffic/CBR]
$cbr set rate_ 10mb
#$cbr set rate_ 100mb
#$cbr set rate_ 1000mb

$cbr set packetSize_ 512
#$cbr set packetSize_ 1024
#$cbr set packetSize_ 2048
$cbr set interval_ 0.5
#$cbr set random_ false
```

```
set ftp [new Application/FTP]
$tcp set window_ 10
#$tcp set window_ 50
#$tcp set window_ 100

$tcp set packetSize_ 512
#$tcp set packetSize_ 1024
#$tcp set packetSize_ 2048
#Attaching the agent to an application(This is to create a connection between
an#application and a sending agent):
$cbr attach-agent $udp
$ftp attach-agent $tcp
#Attaching the agent to a low-level network(This is to create a connection betwe#en each
agent and a node):
$ns attach-agent $node1 $udp
$ns attach-agent $node1 $tcp
#$ns attach-agent $node4 $nullsink
$ns attach-agent $node4 $lostpktsink_udp
$ns attach-agent $node4 $tcpsink
#$ns attach-agent $node4 $lostpktsink_tcp

# Associating a sending agent with a receiving agent.
#$ns connect $udp $nullsink
$ns connect $udp $lostpktsink_udp
$ns connect $tcp $tcpsink
#$ns connect $tcp $lostpktsink_tcp

#traffic sinks nullsink/tcpsink/lostpktsink_udp and writes it to the files
#frputfile_udp/frputfile_tcp/ frputpktlost_udp/ frputpktlost_tcp

proc record {} {
        global ns nullsink tcpsink lostpktsink_udp lostpktsink_tcp frputfile_udp
frputfile_tcp frputpktlost_udp frputpktlost_tcp defaultRNG

        #Get an instance of the simulator
```

```
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
        set time 0.5
        #How many bytes have been received by the traffic sinks?
        #set bwidth_udp [$nullsink set bytes_]
        set bwidth_udp [$lostpktsink_udp set bytes_]
        set bwidth_tcp [$tcpsink set bytes_]
        #set bwidth_tcp [$lostpktsink_tcp set bytes_]



        #How many bytes have been lost?
        set lostpkt_udp [$lostpktsink_udp set nlost_]
        set lostpkt_tcp [$lostpktsink_tcp set nlost_]

        #Get the current time
        set now [$ns now]

        #Calculate the throughput (in MBit/s): and write it to the files
        puts $frputfile_udp "$now [expr $bwidth_udp/$time*8/1000000]"
        puts $frputfile_tcp "$now [expr $bwidth_tcp/$time*8/1000000]"
        puts $frputpktlost_udp "$now [expr $lostpkt_udp/1]"
        #puts $frputpktlost_tcp "$now [expr $lostpkt_tcp/1]"

        #Reset the bytes_ values on the traffic sinks
        #$nullsink set bytes_ 0
        $tcpsink set bytes_ 0
        $lostpktsink_udp set bytes_ 0
        #$lostpktsink_tcp set bytes_ 0
        $lostpktsink_udp set nlost_ 0
        #$lostpktsink_tcp set nlost_ 0

        #Re-schedule the procedure
        $ns at [expr $now+$time] "record"
}
```

```
# http-packmime.tcl
# useful constants
set CLIENT 0
set SERVER 1

# removes all packet headers
remove-all-packet-headers
# adds TCP/IP headers
add-packet-header IP TCP
# use the Heap scheduler
$ns use-scheduler Heap

# SETUP PACKMIME
set rate 15
set pm [new PackMimeHTTP]
$pm set-client $node1
$pm set-server $node4
# new connections per second
$pm set-rate $rate
# use HTTP/1.1
$pm set-http-1.1

# SETUP PACKMIME RANDOM VARIABLES
#global defaultRNG
# create RNGs (appropriate RNG seeds are assigned automatically)
set flowRNG [new RNG]
set reqsizeRNG [new RNG]
set rspsizeRNG [new RNG]

# create RandomVariables
set flow_arrive [new RandomVariable/PackMimeHTTPFlowArrive $rate]
set req_size [new RandomVariable/PackMimeHTTPFileSize $rate $CLIENT]
set rsp_size [new RandomVariable/PackMimeHTTPFileSize $rate $SERVER]
```

```
# assign RNGs to RandomVariables
$flow_arrive use-rng $flowRNG
$req_size use-rng $reqsizeRNG
$rsp_size use-rng $rspsizeRNG

# set PackMime variables
$pm set-flow_arrive $flow_arrive
$pm set-req_size $req_size
$pm set-rsp_size $rsp_size
# record HTTP statistics
$pm set-outfile "http_1.dat"

# Scheduling events
$ns at 0.0 "record"
$ns at 2 "$cbr start"
$ns at 7 "$cbr stop"
$ns at 8 "$ftp start"
$ns at 13 "$ftp stop"
$ns at 14 "$pm start"
$ns at 19 "$pm stop"
$ns at 20 "finish"
$ns run
#########################################################
```

Appendix B: Tcl script for scenario 2

```
##########TCL SCRIPT#####################################
set ns [new Simulator]
set mytracefilescen2 [open out_tracescen2.tr w]
set mynamtracefilescen2 [open out_namtracescen2.nam w]
set frputfile_udp [open out_frputscen2_udp.tr w]
set frputfile_tcp [open out_frputscen2_tcp.tr w]
set frputpktlost_udp [open out_lostpktscen2_udp.tr w]
set frputpktlost_tcp [open out_lostpktscen2_tcp.tr w]
$ns trace-all $mytracefilescen2
$ns namtrace-all $mynamtracefilescen2
#Closing the trace files
proc finish {} {
        global ns frputpktlost_udp frputpktlost_tcp frputfile_udp frputfile_tcp
mytracefilescen2 mynamtracefilescen2
        $ns flush-trace
        close $frputpktlost_udp
        close $frputpktlost_tcp
        close $frputfile_udp
        close $frputfile_tcp
        close $mytracefilescen2
        close $mynamtracefilescen2
        #Call xgraph to display the results
        #exec nam out_namtracescen2.nam &

        #exec xgraph out_frputscen2_udp.tr -t "Udp Throughput" &
        #exec xgraph out_frputscen2_tcp.tr -t "Tcp Throughput" &
        #exec xgraph out_lostpktscen2_udp.tr -t "Udp Lost" &

        exec awk -f delay.awk out_tracescen2.tr > delayfilescen2.dat

        #exec xgraph out_lostpktscen2_tcp.tr -t "Tcp Lost" &
        exit 0
}
```

```
# routing protocol Distance Vector
$ns rtproto DV
#
# make nodes
set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]

# make links
$ns duplex-link $node1 $node2 100Mb 10ms DropTail
$ns duplex-link $node2 $node3 100Mb 10ms DropTail
$ns duplex-link $node3 $node4 100Mb 10ms DropTail

# Creation of sending agent
set udp [new Agent/UDP]
set tcp [new Agent/TCP]

#Creation of receiving agent / Sink
set nullsink [new Agent/Null]
set tcpsink [new Agent/TCPSink]
set lostpktsink_udp [new Agent/LossMonitor]
set lostpktsink_tcp [new Agent/LossMonitor]

# Creation of application (This is a modelling of users demand):
set cbr [new Application/Traffic/CBR]
$cbr set rate_ 10mb
#$cbr set rate_ 100mb
#$cbr set rate_ 1000mb

$cbr set packetSize_ 512
#$cbr set packetSize_ 1024
#$cbr set packetSize_ 2048
$cbr set interval_ 0.5
```

```
#$cbr set random_ false

set ftp [new Application/FTP]
$tcp set window_ 10
#$tcp set window_ 50
#$tcp set window_ 100

$tcp set packetSize_ 512
#$tcp set packetSize_ 1024
#$tcp set packetSize_ 2048
#Attaching the agent to an application(This is to create a connection between
an#application and a sending agent):
$cbr attach-agent $udp
$ftp attach-agent $tcp
#Attaching the agent to a low-level network(This is to create a connection betwe#en each
agent and a node):
$ns attach-agent $node1 $udp
$ns attach-agent $node1 $tcp
#$ns attach-agent $node4 $nullsink
$ns attach-agent $node4 $lostpktsink_udp
$ns attach-agent $node4 $tcpsink
#$ns attach-agent $node4 $lostpktsink_tcp

# Associating a sending agent with a receiving agent.
#$ns connect $udp $nullsink
$ns connect $udp $lostpktsink_udp
$ns connect $tcp $tcpsink
#$ns connect $tcp $lostpktsink_tcp

#traffic sinks nullsink/tcpsink/lostpktsink_udp and writes it to the files
#frputfile_udp/frputfile_tcp/ frputpktlost_udp/ frputpktlost_tcp

proc record {} {
        global ns nullsink tcpsink lostpktsink_udp lostpktsink_tcp frputfile_udp
frputfile_tcp frputpktlost_udp frputpktlost_tcp defaultRNG
```

```
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
        set time 0.5
        #How many bytes have been received by the traffic sinks?
        #set bwidth_udp [$nullsink set bytes_]
        set bwidth_udp [$lostpktsink_udp set bytes_]
        set bwidth_tcp [$tcpsink set bytes_]
        #set bwidth_tcp [$lostpktsink_tcp set bytes_]

        #How many bytes have been lost?
        set lostpkt_udp [$lostpktsink_udp set nlost_]
        set lostpkt_tcp [$lostpktsink_tcp set nlost_]

        #Get the current time
        set now [$ns now]
        #Calculate the throughput (in MBit/s): and write it to the files
        puts $frputfile_udp "$now [expr $bwidth_udp/$time*8/1000000]"
        puts $frputfile_tcp "$now [expr $bwidth_tcp/$time*8/1000000]"
        puts $frputpktlost_udp "$now [expr $lostpkt_udp/1]"
        #puts $frputpktlost_tcp "$now [expr $lostpkt_tcp/1]"

        #Reset the bytes_ values on the traffic sinks
        #$nullsink set bytes_ 0
        $tcpsink set bytes_ 0
        $lostpktsink_udp set bytes_ 0
        #$lostpktsink_tcp set bytes_ 0
        $lostpktsink_udp set nlost_ 0
        #$lostpktsink_tcp set nlost_ 0

        #Re-schedule the procedure
        $ns at [expr $now+$time] "record"
}
# http-packmime.tcl
```

```
# useful constants
set CLIENT 0
set SERVER 1

# removes all packet headers
remove-all-packet-headers
# adds TCP/IP headers
add-packet-header IP TCP
# use the Heap scheduler
$ns use-scheduler Heap
# SETUP PACKMIME
set rate 15
set pm [new PackMimeHTTP]
$pm set-client $node1
$pm set-server $node4
# new connections per second
$pm set-rate $rate
# use HTTP/1.1
$pm set-http-1.1

# SETUP PACKMIME RANDOM VARIABLES
#global defaultRNG

# create RNGs (appropriate RNG seeds are assigned automatically)
set flowRNG [new RNG]
set reqsizeRNG [new RNG]
set rspsizeRNG [new RNG]

# create RandomVariables
set flow_arrive [new RandomVariable/PackMimeHTTPFlowArrive $rate]
set req_size [new RandomVariable/PackMimeHTTPFileSize $rate $CLIENT]
set rsp_size [new RandomVariable/PackMimeHTTPFileSize $rate $SERVER]
# assign RNGs to RandomVariables
$flow_arrive use-rng $flowRNG
$req_size use-rng $reqsizeRNG
```

```
$rsp_size use-rng $rspsizeRNG

# set PackMime variables
$pm set-flow_arrive $flow_arrive
$pm set-req_size $req_size
$pm set-rsp_size $rsp_size

# record HTTP statistics
$pm set-outfile "http_2.dat"
# Scheduling events
$ns at 0.0 "record"
$ns at 2 "$cbr start"
$ns at 7 "$cbr stop"
$ns at 8 "$ftp start"
$ns at 13 "$ftp stop"
$ns at 14 "$pm start"
$ns at 19 "$pm stop"
$ns at 20 "finish"
$ns run
#########################################################
```

Appendix C: Awk script for delay

```
##########AWK SCRIPT#################################
BEGIN {
# simple awk script to generate end-to-end packet lifetime statistics
# in a form suitable for plotting with xgraph.
# Lloyd Wood, July 1999.
# http://www.ee.surrey.ac.uk/Personal/L.Wood/ns/
highest_packet_id = 0;
}
{
action = $1;
time = $2;
node_1 = $3;
node_2 = $4;
src = $5;
flow_id = $8;
node_1_address = $9;
node_2_address = $10;
seq_no = $11;
packet_id = $12;
if ( packet_id > highest_packet_id ) highest_packet_id = packet_id;

# getting start time is not a problem, provided you're not starting
# traffic at 0.0.
# could test for sending node_1_address or flow_id here.
if ( start_time[packet_id] == 0 ) start_time[packet_id] = time;

# only useful for small unicast where packet_id doesn't wrap.
# checking receive means avoiding recording drops
if ( action != "d" ) {
if ( action == "r" ) {
# could test for receiving node_2_address or flow_id here.
end_time[packet_id] = time;
```

```
}
} else {
end_time[packet_id] = -1;
}
}
END {
for ( packet_id = 0; packet_id <= highest_packet_id; packet_id++ ) {
start = start_time[packet_id];
end = end_time[packet_id];
packet_duration = end - start;
if ( start < end ) printf("%d %f\n", start, packet_duration);
}
}
##############################################
```

Appendix D: HTTP throughput

Connection rate 5

			No VPN				With VPN
427	10410	220.45	49.15854	427	10410	460.974	23.50892
479	182	57.414	11.51287	479	182	137.545	4.8057
571	1167	82.703	21.01496	510	134	195.826	3.288634
510	134	115.697	5.566264	571	1167	242.946	7.153853
874	1375	180.212	12.47975	874	1375	340.537	6.604275
479	182	43.033	15.3603	479	182	83.151	7.949393
3804	14653	186.299	99.07192	470	1202	182.587	9.157278
479	182	22.267	29.68518	479	182	160.178	4.126659
470	1202	160.159	10.43963	3804	14653	547.052	33.73902
733	6066	122.694	55.41428	441	419	220.248	3.904689
441	419	140.085	6.13913	733	6066	363.173	18.72111
733	2895	74.925	48.42176	317	630	187.309	5.055817
733	2940	63.782	57.58678	386	1028	199.524	7.086867
431	7376	119.631	65.259	733	2895	244.11	14.86215
317	630	65.683	14.41773	431	7376	315.566	24.73967
386	1028	67.156	21.05545	733	2940	246.25	14.91574
474	7764	101.483	81.17616	474	7764	301.922	27.28519
869	175	61.439	16.99246	869	175	181.645	5.747474
445	248	140.071	4.947491	445	248	220.208	3.147025
417	17562	141.258	127.2777	343	493	174.615	4.787676
1013	6731	123.979	62.46219	417	17562	421.908	42.61356
343	493	60.427	13.83488	1013	6731	364.523	21.2442
417	701	94.475	11.83382	417	701	140.651	7.948753
417	572	47.483	20.82851	441	609	187.704	5.593914
441	609	62.722	16.74054	417	572	122.892	8.047717
278	938	160.12	7.594304	278	938	280.353	4.337389
417	775	43.328	27.51108	491	594	189.24	5.73346
491	594	63.204	17.16664	396	2647	243.819	12.48057
396	2647	89.152	34.13272	452	1944	243.621	9.834949
452	1944	83.112	28.82857	447	3460	253.813	15.39322
447	3460	82.579	47.31227	488	5168	303.711	18.62297
488	5168	113.494	49.83523	428	7503	303.264	26.15213
428	7503	103.172	76.87163	447	4898	123.315	43.34428
447	4898	46.955	113.8324	417	775	126.813	9.399667
447	11330	87.254	134.9738	488	125	128.237	4.780212
447	3576	48.399	83.12155	467	202	120.422	5.555463
488	125	42.091	14.56368	447	11330	207.547	56.74377
447	9922	41.117	252.1828	418	133	122.08	4.513434
467	202	41.975	15.93806	443	329	121.185	6.370425

860	482	61.511	21.81724	409	185	127.623	4.654333
418	133	96.299	5.721762	860	482	280.383	4.78631
443	329	47.517	16.24682	447	3576	121.973	32.98271
831	2665	200.275	17.456	831	2665	356.966	9.79365
448	123269	322.571	383.5342	447	9922	221.123	46.89245
409	185	140.064	4.240918	448	123269	965.201	128.1774
831	9174	217.546	45.99027	831	9174	458.108	21.83983
831	3172	82.283	48.64917				

Connection rate 10

			No VPN				**With VPN**
427	10410	220.45	49.15854	494	134	122.724	5.117173
494	134	42.598	14.74248	478	182	137.545	4.79843
478	182	57.414	11.49545	427	10410	460.974	23.50892
534	1167	135.792	12.52651	534	1167	256.016	6.644116
860	1375	180.211	12.40213	860	1375	340.534	6.563221
3492	14653	184.013	98.60716	478	182	62.385	10.57947
478	182	22.267	29.64027	461	1202	185.744	8.953183
461	1202	65.525	25.37963	3492	14653	504.703	35.95184
478	182	22.327	29.56062	438	419	220.247	3.891086
438	419	140.085	6.117714	478	182	160.178	4.120416
743	6066	220.428	30.88991	743	6066	363.175	18.74854
377	1028	63.591	22.09432	377	1028	183.854	7.641933
312	630	79.371	11.86831	312	630	199.41	4.723936
427	7376	115.156	67.76026	427	7376	315.565	24.72708
743	2895	87.301	41.67192	473	116	125.768	4.683226
743	2940	65.912	55.87753	743	2895	247.677	14.68849
473	116	41.001	14.3655	743	2940	181.549	20.28653
823	20152	161.97	129.4993	416	168	121.065	4.823855
416	168	40.946	14.26269	432	242	220.203	3.060812
416	168	23.527	24.82254	416	168	63.633	9.177628
432	242	140.07	4.81188	823	20152	482.684	43.45493
416	168	27.444	21.2797	1167	105	200.628	6.340092
1167	105	114.5	11.10917	416	168	67.55	8.645448
337	938	80.469	15.84461	337	938	234.739	5.431564
448	2647	82.853	37.35532	448	2647	243.227	12.72474
283	594	160.093	5.478066	283	594	280.271	3.129114
405	1944	83.394	28.16749	493	112	123.121	4.913865
493	112	42.998	14.07042	405	1944	243.735	9.637516
448	3460	83.403	46.85683	448	3460	243.736	16.03374
492	5168	109.242	51.81157	492	5168	309.697	18.27593
452	9474	133.542	74.32868	452	9474	374.092	26.53358
428	13717	123.005	114.9953	416	168	62.814	9.29729

416	168	22.708	25.71781	428	13717	363.544	38.90863
452	701	46.594	24.74568	410	482	120.476	7.403964
452	572	48.143	21.26997	445	132	128.228	4.499797
445	132	42.088	13.70937	768	329	182.23	6.019865
410	482	41.993	21.24164	433	554	182.324	5.41344
768	329	61.068	17.96358	452	701	121.273	9.507475
454	4885	100.517	53.11539	829	185	236.543	4.286747
387	119	41.438	12.21101	454	4885	327.762	16.28926
433	554	116.34	8.483755	387	119	220.163	2.298297
452	775	47.55	25.80442	452	572	121.666	8.416484
829	185	160.104	6.333383	1079	256	181.216	7.366899
474	123269	348.677	354.8929	429	2691	247.974	12.58196
452	1386	140.163	13.1133	403	1100	182.445	8.238099
429	2691	82.216	37.94882	433	1671	340.456	6.179947
433	1671	81.072	25.95224	768	10856	363.051	32.01754
1079	256	99.74	13.3848	676	3011	310.489	11.87482
768	10856	217.685	53.39826	452	775	220.333	5.568844
450	22862	190.035	122.6721	285	4555	340.272	14.22391
403	1100	61.017	24.63248	450	22862	518.652	44.94729
676	3011	110.092	33.49017	449	27103	471.039	58.49197
449	27103	143.305	192.2613	452	1386	121.268	15.15651
285	4555	200.254	24.1693	474	123269	969.679	127.6123
768	3861	101.68	45.52518	428	4196	302.144	15.30396
11838	1665	260.642	51.80669	768	3861	302.643	15.29525
428	4196	102.184	45.2517	11838	1665	581.276	23.22993
321	721	62.716	16.61458	321	721	182.921	5.696448
563	2889	100.668	34.29094	276	309	121.828	4.801852
276	309	41.709	14.02575	563	2889	301.027	11.46741
2125	174	82.494	27.86869	2125	174	244.064	9.41966
904	720	83.719	19.39823	904	720	242.794	6.688798
439	142	46.522	12.48872	439	142	126.64	4.587808
352	10524	142.498	76.32388	277	576	183.663	4.644376
1649	463	83.552	25.27767	448	67	220.165	2.339155
448	67	140.057	3.677074	352	10524	383.028	28.39479
277	576	160.091	5.32822	1649	463	340.458	6.203408
1649	3138	102.366	46.76357	277	3129	182.52	18.66097
277	3129	61.362	55.50667	364	1455	181.614	10.01575
364	1455	61.76	29.45272	552	1013	242.044	6.465767
552	1013	126.602	12.36157	1649	3138	287.215	16.66696
1649	1383	81.27	37.30774	893	109	182.302	5.496374
277	6883	102.481	69.86661	275	1326	181.404	8.825605
275	1326	62.154	25.7586	277	6883	262.906	27.23407
893	109	93.084	10.76447	372	5303	333.813	17.00054
655	6262	135.314	51.11814	655	6262	375.812	18.40548

351	7117	107.405	69.53121	1649	1383	242.891	12.48297
372	5303	200.324	28.32911	351	7117	400.882	18.62892
826	19455	162.876	124.518	826	19455	488.191	41.54317
373	578	41.255	23.05175	298	990	190.185	6.772353
431	773	60.639	19.85521	948	23801	489.438	50.56616
407	892	69.941	18.5728	373	578	126.553	7.514638
298	990	66.412	19.39409	398	245231	1784.324	137.6594

Connection rate 15

			No VPN				**With VPN**
487	134	42.597	14.57849	487	134	122.722	5.060217
476	182	57.414	11.46062	476	182	137.545	4.783889
427	10410	220.45	49.15854	517	1167	256.012	6.577817
517	1167	135.79	12.4015	427	10410	460.974	23.50892
851	1375	180.21	12.35226	851	1375	340.532	6.536831
456	1202	65.525	25.30332	456	1202	203.35	8.15343
3301	14653	183.998	97.57715	436	419	122.455	6.982157
476	182	22.267	29.55046	476	182	160.177	4.107956
436	419	42.352	20.18795	3301	14653	487.051	36.86267
476	182	39.183	16.793	310	630	183.743	5.115841
310	630	63.554	14.79057	373	1028	195	7.184615
746	6066	220.428	30.90351	746	6066	363.176	18.75675
373	1028	74.737	18.74574	476	182	79.301	8.297499
424	7376	200.507	38.90138	472	116	127.203	4.622532
746	2895	87.301	41.70628	424	7376	400.961	19.45326
472	116	45.648	12.88118	746	2895	246.242	14.78627
746	2940	61.265	60.16486	427	242	121.536	5.504542
789	20152	161.967	129.2918	746	2940	181.495	20.3091
416	168	40.946	14.26269	416	168	220.182	2.652351
416	168	23.527	24.82254	334	938	183.84	6.91906
427	242	140.07	4.776183	1278	105	234.683	5.893056
334	938	80.469	15.80733	416	168	67.55	8.645448
1278	105	114.509	12.07765	789	20152	482.281	43.42074
416	168	27.444	21.2797	287	594	182.888	4.817156
452	2647	82.853	37.40359	452	2647	260.957	11.87552
287	594	160.093	5.503051	416	168	160.159	3.646376
495	112	42.998	14.11694	495	112	123.121	4.930109
410	1944	83.394	28.22745	410	1944	243.736	9.657991
446	3460	83.403	46.83285	446	3460	243.736	16.02554
492	5168	109.242	51.81157	492	5168	309.697	18.27593
455	9474	133.542	74.35114	406	482	128.556	6.907496
428	13717	123.005	114.9953	443	132	147.241	3.905162
455	701	42.76	27.03461	455	9474	374.092	26.5416

406	482	40.307	22.03091	428	13717	363.544	38.90863
443	132	48.11	11.95178	727	329	182.658	5.781296
727	329	62.133	16.9958	808	185	181.257	5.478409
808	185	61.06	16.26269	377	119	122.066	4.063376
429	554	62.007	15.85305	429	554	180.517	5.445471
449	4885	127.358	41.88194	449	4885	307.188	17.36396
416	168	21.435	27.24516	455	701	176.554	6.547572
455	572	96.333	10.66094	431	1671	241.837	8.691805
377	119	140.056	3.541441	403	1100	181.257	8.292094
425	2691	87.681	35.53792	1043	256	280.373	4.633114
1043	256	61.044	21.27973	425	2691	340.534	9.150334
467	123269	328.117	377.1094	455	572	122.494	8.384084
727	10856	122.483	94.56823	283	4555	302.637	15.98615
455	775	89.313	13.77179	678	3011	340.299	10.84047
403	1100	69.944	21.48862	449	22862	428.919	54.34826
431	1671	180.154	11.6678	416	168	160.159	3.646376
283	4555	102.36	47.26456	727	10856	410.269	28.2327
449	22862	238.005	97.94332	455	775	121.204	10.14818
455	1386	41.034	44.86523	442	27103	519.007	53.0725
442	27103	180.806	152.3456	424	4196	310.492	14.87961
678	3011	200.294	18.41793	12280	1665	482.832	28.88168
727	3861	101.677	45.12328	273	309	120.575	4.826871
424	4196	102.184	45.21256	467	123269	965.88	128.107
12280	1665	260.678	53.49512	455	1386	122.348	15.04724
318	721	62.715	16.56701	318	721	280.31	3.706611
553	2889	100.668	34.1916	553	2889	303.241	11.35071
273	309	41.709	13.95382	2088	174	244.055	9.268403
2088	174	83.727	27.01637	887	720	242.112	6.637424
887	720	82.481	19.48327	434	142	126.639	4.548362
434	142	46.521	12.38151	727	3861	303.006	15.14161
344	10524	142.497	76.26827	265	576	183.66	4.579114
439	67	43.454	11.6445	439	67	220.163	2.298297
1515	463	160.184	12.3483	1515	463	280.453	7.052875
265	576	160.09	5.253295	344	10524	383.026	28.37405
1515	3138	82.395	56.47187	356	1455	181.612	9.971808
265	3129	61.361	55.31201	265	3129	182.517	18.59553
540	1013	81.742	18.9988	540	1013	286.901	5.413017
356	1455	106.619	16.98572	1515	3138	242.36	19.19871
262	1326	61.194	25.95026	262	1326	181.401	8.754086
876	109	82.139	11.99187	876	109	202.335	4.868164
1515	1383	62.258	46.54823	371	5303	302.782	18.73956
346	7117	115.358	64.69426	334	805	187.259	6.082485
371	5303	133.305	42.56404	346	7117	333.861	22.35361
265	6883	87.366	81.81673	670	1124	244.802	7.328372

651	6262	220.436	31.36058	265	6883	243.163	29.39592
334	805	81.281	14.01311	651	6262	460.933	14.99784
828	19455	162.876	124.5303	369	2378	261.695	10.49695
670	1124	84.55	21.21821	1515	1383	235.924	12.28362
369	2378	135.545	20.26633	828	19455	496.145	40.88119
427	688	60.494	18.43158	427	688	280.328	3.977484
346	7160	107.148	70.05264	346	7160	301.273	24.91428
427	559	48.115	20.49257	1436	4995	367.564	17.49627
1436	4995	220.394	29.17956	427	559	121.177	8.136858
427	682	41.003	27.0468	407	2150	248.465	10.29119
407	2150	80.519	31.75648	278	150	121.349	3.527017
886	4182	138.827	36.50587	443	996	198.957	7.232719
2871	2154	124.873	40.24088	377	3652	245.177	16.43303
377	3652	81.337	49.53465	886	4182	361.146	14.03311
278	150	41.255	10.3745	2871	2154	361.839	13.88739
443	996	160.138	8.986	427	682	220.305	5.03393
877	3865	220.249	21.53018	877	3865	460.746	10.292
425	21202	141.216	153.1484	319	370	122.202	5.638206
390	18715	145.743	131.0869	425	21202	421.85	51.26704
2871	12265	152.817	99.04657	390	18715	426.403	44.80503
425	2440	83.914	34.1421	347	9829	372.945	27.28552
347	9829	220.394	46.17186	425	2440	340.556	8.412713
319	370	41.903	16.44274	498	138	122.857	5.17675
443	3453	82.302	47.33785	443	3453	244.334	15.94539
469	1311	61.213	29.07879	469	1311	187.007	9.51836
498	138	42.729	14.8845	466	142	121.63	4.998767
428	11631	127.155	94.83701	434	707	182.524	6.251233
1041	7664	120.87	72.01953	943	142	185.733	5.841719
434	707	62.303	18.31372	1799	104	241.421	7.882496
943	142	65.055	16.6782	428	11631	363.102	33.21105
1799	104	85.55	22.2443	2871	12265	422.525	35.82273
466	142	41.49	14.65413	434	2018	245.486	9.98835
426	16577	214.507	79.26548	1041	7664	361.396	24.08715
434	2018	81.708	30.0093	247	207	122.845	3.695714
331	11370	124.462	94.01263	390	2493	241.986	11.91391
390	2493	81.235	35.48963	426	16577	455.075	37.36307
196	4348	101.419	44.80423	331	11370	364.953	32.06166
684	268	62.793	15.16093	378	463	129.554	6.491502
398	245231	661.628	371.2494	684	268	205.689	4.628347
378	463	41.002	20.51119	418	459	121.171	7.237705
247	207	65.452	6.936381	429	600	180.619	5.697075
714	37844	186.192	207.0873	196	4348	301.782	15.05723
429	600	69.415	14.82389	1411	466	183.814	10.21141
418	459	47.846	18.32964	447	1544	185.694	10.72194

1411	466	66.096	28.39809	528	639	186.261	6.265402
447	1544	65.423	30.43272	373	7261	305.637	24.97734
948	23801	161.272	153.4612	431	773	181.627	6.62897
528	639	63.501	18.37766	407	892	180.893	7.181041
373	7261	105.209	72.56033	409	10496	362.977	30.04323
409	10496	122.442	89.06258	714	37844	546.988	70.49149
373	578	41.255	23.05175	298	990	190.185	6.772353
431	773	60.639	19.85521	948	23801	489.438	50.56616
407	892	69.941	18.5728	373	578	126.553	7.514638
298	990	66.412	19.39409	398	245231	1784.324	137.6594

Appendix E: Instant CBR throughput

CBR transfer rate: 10 Mb								
Packet size: 512 Bytes			Packet size: 1024 Bytes			Packet size: 2048 Bytes		
Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn
2	0	0	2	0	0	2	0	0
2.5	0.008	0.008	2.5	0.016	0.016	2.5	0.033	0.033
3	0.008	0.008	3	0.016	0.016	3	0.033	0.033
3.5	0.008	0.008	3.5	0.016	0.016	3.5	0.033	0.033
4	0.008	0.008	4	0.016	0.016	4	0.033	0.033
4.5	0.008	0.008	4.5	0.016	0.016	4.5	0.033	0.033
5	0.008	0.008	5	0.016	0.016	5	0.033	0.033
5.5	0.008	0.008	5.5	0.016	0.016	5.5	0.033	0.033
6	0.008	0.008	6	0.016	0.016	6	0.033	0.033
6.5	0.008	0.008	6.5	0.016	0.016	6.5	0.033	0.033
7	0.008	0.008	7	0.016	0.016	7	0.033	0.033
CBR transfer rate: 100 Mb								
Packet size: 512 Bytes			Packet size: 1024 Bytes			Packet size: 2048 Bytes		
Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn
2	0	0	2	0	0	2	0	0
2.5	0.008	0.008	2.5	0.016	0.016	2.5	0.033	0.033
3	0.008	0.008	3	0.016	0.016	3	0.033	0.033
3.5	0.008	0.008	3.5	0.016	0.016	3.5	0.033	0.033
4	0.008	0.008	4	0.016	0.016	4	0.033	0.033
4.5	0.008	0.008	4.5	0.016	0.016	4.5	0.033	0.033
5	0.008	0.008	5	0.016	0.016	5	0.033	0.033
5.5	0.008	0.008	5.5	0.016	0.016	5.5	0.033	0.033
6	0.008	0.008	6	0.016	0.016	6	0.033	0.033
6.5	0.008	0.008	6.5	0.016	0.016	6.5	0.033	0.033
7	0.008	0.008	7	0.016	0.016	7	0.033	0.033
CBR transfer rate: 1000 Mb								
Packet size: 512 Bytes			Packet size: 1024 Bytes			Packet size: 2048 Bytes		
Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn
2	0	0	2	0	0	2	0	0
2.5	0.008	0.008	2.5	0.016	0.016	2.5	0.033	0.033
3	0.008	0.008	3	0.016	0.016	3	0.033	0.033
3.5	0.008	0.008	3.5	0.016	0.016	3.5	0.033	0.033
4	0.008	0.008	4	0.016	0.016	4	0.033	0.033
4.5	0.008	0.008	4.5	0.016	0.016	4.5	0.033	0.033
5	0.008	0.008	5	0.016	0.016	5	0.033	0.033
5.5	0.008	0.008	5.5	0.016	0.016	5.5	0.033	0.033
6	0.008	0.008	6	0.016	0.016	6	0.033	0.033
6.5	0.008	0.008	6.5	0.016	0.016	6.5	0.033	0.033
7	0.008	0.008	7	0.016	0.016	7	0.033	0.033

Appendix F: Instant FTP throughput

FTP window size: 10 Kb									
Packet size: 512			Packet size: 1024			Packet size: 2048		
Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn
8	0	0	8	0	0	8	0	0
8.5	1.979	0.48	8.5	3.81	0.92	8.5	7.48	1.8
9	2.208	0.79	9	4.26	1.53	9	8.35	3.01
9.5	2.208	0.71	9.5	4.26	1.36	9.5	8.02	2.67
10	2.208	0.71	10	4.26	1.36	10	8.35	2.67
10.5	2.208	0.79	10.5	4.09	1.36	10.5	8.35	2.67
11	2.208	0.71	11	4.26	1.53	11	8.35	3.01
11.5	2.208	0.71	11.5	4.26	1.36	11.5	8.15	2.67
12	2.208	0.79	12	4.26	1.36	12	8.22	2.67
12.5	2.12	0.71	12.5	4.26	1.53	12.5	8.35	2.67
13	0.088	0.71	13	4.26	1.36	13	8.35	3.01
FTP window size: 50 Kb								
Packet size: 512			Packet size: 1024			Packet size: 2048		
Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn
8	0	0	8	0	0	8	0	0
8.5	8.94	1.43	8.5	17.23	2.76	8.5	33.34	5.41
9	11.04	3.97	9	21.28	7.66	9	40.89	13.73
9.5	11.04	3.53	9.5	21.13	6.81	9.5	41.43	14.67
10	11.04	3.53	10	20.84	6.81	10	41.76	13.36
10.5	11.04	3.97	10.5	21.02	6.81	10.5	41.76	13.36
11	11.04	3.53	11	21.28	7.66	11	41.06	14.37
11.5	10.95	3.53	11.5	21.28	6.81	11.5	40.92	14.03
12	10.8	3.65	12	21.28	6.81	12	41.63	13.36
12.5	10.93	3.85	12.5	21.28	7.66	12.5	41.76	13.36
13	10.04	3.53	13	21.28	6.81	13	41.69	15
FTP window size: 100 Kb								
Packet size: 512			Packet size: 1024			Packet size: 2048		
Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn	Time	WithoutVpn	WithVpn
8	0	0	8	0	0	8	0	0
8.5	17.01	1.997	8.5	32.67	3.85	8.5	62.21	7.55
9	22.08	7.95	9	42.12	14.78	9	82.65	27.09
9.5	22.08	7.07	9.5	42.12	14.16	9.5	83.19	29.7
10	22.08	7.07	10	42.12	13.62	10	82.85	26.73
10.5	22.03	7.9	10.5	42.3	13.62	10.5	82.68	26.73
11	21.84	7.11	11	42.56	15.32	11	82.65	27.73
11.5	21.84	7.07	11.5	42.56	13.62	11.5	82.68	29.06
12	21.84	7.19	12	42.56	13.62	12	83.35	26.73
12.5	21.97	7.83	12.5	42.56	15.32	12.5	82.68	26.73
13	22.08	7.07	13	42.49	13.62	13	82.65	28.36